// HALBORN

DeBridge - DLN EVM Contracts

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **December 13th, 2022 - December 27th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	12/13/2022	Roberto Reigada
0.2	Document Updates	12/26/2022	Roberto Reigada
0.3	Draft Review	12/28/2022	Piotr Cielas
0.4	Draft Review	12/28/2022	Gabi Urrutia
1.0	Remediation Plan	01/26/2023	Roberto Reigada
1.1	Remediation Plan Review	01/26/2023	Piotr Cielas
1.2	Remediation Plan Review	01/26/2023	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Piotr Cielas	Halborn	Piotr.Cielas@halborn.com
Roberto Reigada	Halborn	Roberto.Reigada@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

DeBridge is a cross-chain interoperability and liquidity transfer protocol that allows truly decentralized transfer of assets between various blockchains.

DeBridge engaged Halborn to conduct a security audit on their smart contracts beginning on December 13th, 2022 and ending on December 27th, 2022. The security assessment was scoped to the smart contracts provided to the Halborn team.

1.2 AUDIT SUMMARY

The team at Halborn was provided two weeks for the engagement and assigned a full-time security engineer to audit the security of the smart contract. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that smart contract functions operate as intended
- Identify potential security issues with the smart contracts

In summary, Halborn identified some recommendations that were mostly addressed by the DeBridge team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing

techniques help enhance coverage of the contracts' solidity code and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

- Research into architecture and purpose.
- Smart contract manual code review and walkthrough.
- Manual assessment of use and safety for the critical Solidity variables and functions in scope to identify any arithmetic related vulnerability classes.
- Manual testing by custom scripts. (Hardhat).
- Static Analysis of security for scoped contract, and imported functions manually.
- Testnet deployment (Ganache).

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.

4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

The security assessment was scoped to the following smart contracts on the main branch for the audit:

- DlnDestination.sol
- DlnSource.sol
- DlnBase.sol

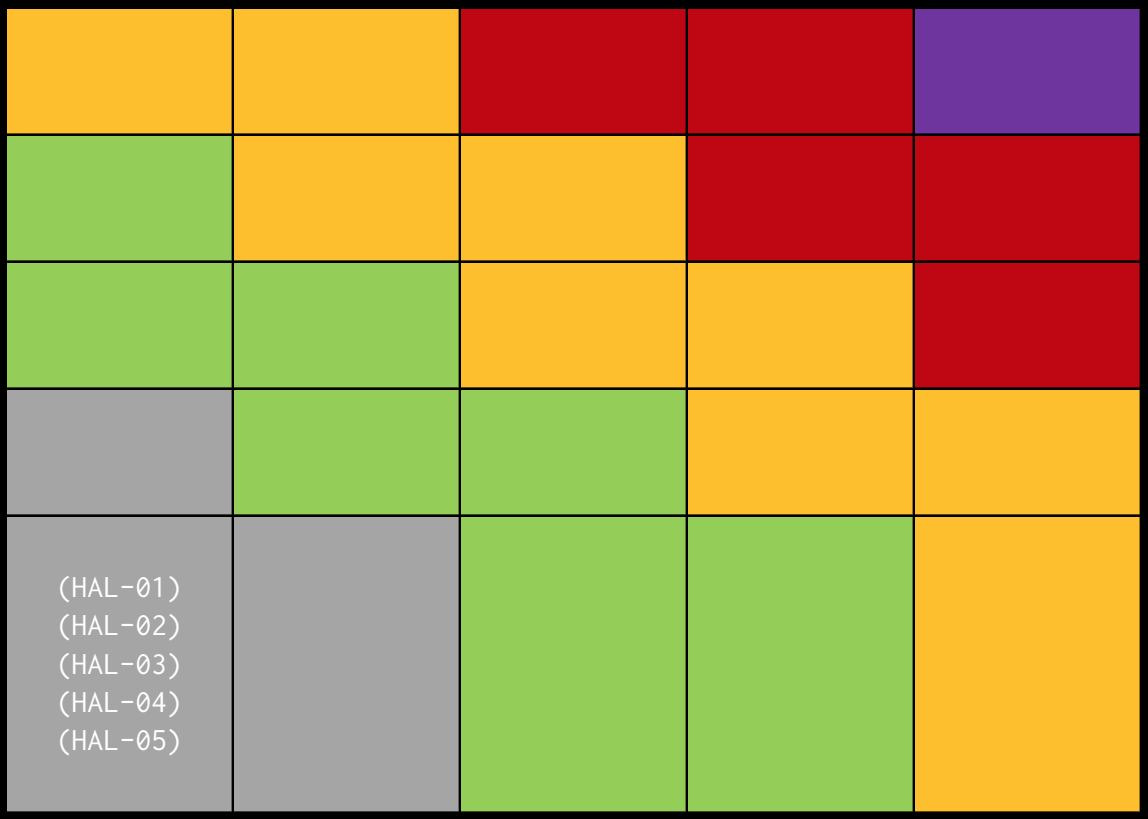
Commit ID: c129630576f69ab49e4e1bcf9ed1a97948975a94

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	0	5

LIKELIHOOD

IMPACT



SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - REDUNDANT ORDERCREATION.EXTERNALCALL ARRAY LENGTH CHECK	Informational	FUTURE RELEASE
HAL02 - GIVEORDERSTATE STRUCT CAN BE PACKED	Informational	ACKNOWLEDGED
HAL03 - UNUSED DECLARED CUSTOM ERRORS	Informational	SOLVED - 01/26/2023
HAL04 - UNUSED CONSTANT STATE VARIABLE	Informational	SOLVED - 01/26/2023
HAL05 - UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0	Informational	SOLVED - 01/26/2023

FINDINGS & TECH DETAILS

3.1 (HAL-01) REDUNDANT ORDERCREATION.EXTERNALCALL ARRAY LENGTH CHECK - INFORMATIONAL

Description:

In the DlnSource smart contract, the function createOrder() blocks all external call during order creation and reverts if an external call is detected:

Listing 1: DlnSource.sol (Line 183)

```
170 /// @dev Create new order for DLN system
171 /// @param _orderCreation Input parameters for new order
172 /// @param _affiliateFee Affiliate amount that will be paid for
 ↳ beneficiary. bytes (beneficiary + affiliate amount)
173 /// @param _referralCode referral code, just emit in event
174 /// @param _permitEnvelope Permit for approving the spender by
 ↳ signature.bytes (amount + deadline + signature)
175 function createOrder(
176     OrderCreation calldata _orderCreation,
177     bytes calldata _affiliateFee,
178     uint32 _referralCode,
179     bytes calldata _permitEnvelope
180 ) external payable nonReentrant whenNotPaused returns (bytes32) {
181
182
183     if (_orderCreation.externalCall.length > 0) revert
 ↳ ExternalCallIsBlocked();
184
185     uint256 affiliateAmount;
186     address affiliateBeneficiary;
187     if (_affiliateFee.length > 0) {
188         if (_affiliateFee.length != 52) revert
 ↳ WrongAffiliateFeeLength();
189         affiliateBeneficiary = BytesLib.toAddress(_affiliateFee,
 ↳ 0);
190         affiliateAmount = BytesLib.toUint256(_affiliateFee, 20);
191         if (affiliateAmount > 0 && affiliateBeneficiary == address
 ↳ (0)) revert ZeroAddress();
192     }
```

```
193
194     Order memory _order = validateCreationOrder(_orderCreation,
 ↳ msg.sender);
195
196     if (_orderCreation.giveTokenAddress == address(0)) {
197         if (msg.value != _order.giveAmount + globalFixedNativeFee)
 ↳  revert MismatchNativeGiveAmount();
198     }
199     else
200     {
201         if (msg.value != globalFixedNativeFee) revert
 ↳ WrongFixedFee(msg.value, globalFixedNativeFee);
202
203         _executePermit(_orderCreation.giveTokenAddress,
 ↳ _permitEnvelope);
204         _safeTransferFrom(
205             _orderCreation.giveTokenAddress,
206             msg.sender,
207             address(this),
208             _order.giveAmount
209         );
210     }
211     // reduce giveAmount on (percentFee + affiliateFee)
212     uint256 percentFee = (globalTransferFeeBps * _order.giveAmount
 ↳ ) / BPS_DENOMINATOR;
213     _order.giveAmount -= percentFee + affiliateAmount;
214
215     bytes32 orderId = getOrderId(_order);
216     {
217         GiveOrderState storage orderState = giveOrders[orderId];
218         orderState.status = OrderGiveStatus.Created;
219         orderState.giveTokenAddress =  uint160(_orderCreation.
 ↳ giveTokenAddress);
220         orderState.nativeFixFee = globalFixedNativeFee;
221         orderState.takeChainId = _order.takeChainId.toUint48();
222         orderState.percentFee = percentFee.toUint208();
223         orderState.giveAmount = _order.giveAmount;
224         // save affiliate_fee to storage
225         if (affiliateAmount > 0) {
226             orderState.affiliateAmount = affiliateAmount;
227             orderState.affiliateBeneficiary = affiliateBeneficiary
 ↳ ;
228         }
229     }
```

```
230     emit CreatedOrder(
231         _order,
232         orderId,
233         _affiliateFee,
234         globalFixedNativeFee,
235         percentFee,
236         _referralCode
237     );
238     // Increment user nonces
239     masterNonce[msg.sender]++;
240
241     return orderId;
242 }
```

There is no alternative way to place an order in an external call. A few lines below that expression, order params are validated with the validateCreationOrder() function, which again checks for a presence of at least one external call. Because this always evaluates as true, gas is spent unnecessarily and contract security is not improved in any way:

Listing 2: DlnSource.sol (Lines 412-422)

```
390 /// @dev Validate creation of order. Will throw exception if
 ↳ incorrect params was passed
391 /// @param _orderCreation information about order
392 /// @param _sender Sender who create order
393 function validateCreationOrder(OrderCreation memory _orderCreation
 ↳ , address _sender)
394     public
395     view
396     returns (Order memory order)
397 {
398     uint256 dstAddressLength = dlnDestinationAddresses[
 ↳ _orderCreation.takeChainId].length;
399
400     if (dstAddressLength == 0) revert NotSupportedDstChain();
401     if (
402         _orderCreation.takeTokenAddress.length != dstAddressLength
 ↳  ||
403         _orderCreation.receiverDst.length != dstAddressLength ||
404         _orderCreation.orderAuthorityAddressDst.length !=
 ↳ dstAddressLength ||
```

```
405         (_orderCreation.allowedTakerDst.length > 0 &&
406             _orderCreation.allowedTakerDst.length !=
 ↳ dstAddressLength) ||
407         (_orderCreation.allowedCancelBeneficiarySrc.length > 0 &&
408             _orderCreation.allowedCancelBeneficiarySrc.length !=
 ↳ EVM_ADDRESS_LENGTH)
409     ) revert WrongAddressLength();
410
411     // Validate external call params
412     if (_orderCreation.externalCall.length > 0) {
413         ExternalCall memory externalCall = abi.decode(
414             _orderCreation.externalCall,
415             (ExternalCall)
416         );
417         if (externalCall.executionFee > _orderCreation.takeAmount)
 ↳  revert ProposedFeeTooHigh();
418         if (
419             externalCall.data.length > 0 &&
420             externalCall.fallbackAddress.length !=
 ↳ dstAddressLength
421         ) revert WrongAddressLength();
422     }
423     order.giveChainId = getChainId();
424     order.makerOrderNonce = uint64(masterNonce[_sender]);
425     order.makerSrc = abi.encodePacked(_sender);
426     order.giveTokenAddress = abi.encodePacked(_orderCreation.
 ↳ giveTokenAddress);
427     order.giveAmount = _orderCreation.giveAmount;
428     order.takeTokenAddress = _orderCreation.takeTokenAddress;
429     order.takeAmount = _orderCreation.takeAmount;
430     order.takeChainId = _orderCreation.takeChainId;
431     order.receiverDst = _orderCreation.receiverDst;
432     order.givePatchAuthoritySrc = abi.encodePacked(_orderCreation.
 ↳ givePatchAuthoritySrc);
433     order.orderAuthorityAddressDst = _orderCreation.
 ↳ orderAuthorityAddressDst;
434     order.allowedTakerDst = _orderCreation.allowedTakerDst;
435     order.externalCall = _orderCreation.externalCall;
436     order.allowedCancelBeneficiarySrc = _orderCreation.
 ↳ allowedCancelBeneficiarySrc;
437 }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to remove the `if (_orderCreation.externalCall.length > 0)` from the `validateCreationOrder()` function body.

Remediation Plan:

PENDING: The DeBridge team acknowledged this recommendation as the external call layout will be developed in a future release that will use that code.

3.2 (HAL-02) GIVEORDERSTATE STRUCT CAN BE PACKED - INFORMATIONAL

Description:

In the DlnSource smart contract the struct GiveOrderState is defined as given below:

Listing 3: DlnSource.sol (Lines 92,94)

```
86 struct GiveOrderState {
87     OrderGiveStatus status;
88     // stot optimisation
89     uint160 giveTokenAddress;
90     uint88 nativeFixFee;
91     uint48 takeChainId;
92     uint208 percentFee;
93     uint256 giveAmount;
94     address affiliateBeneficiary;
95     uint256 affiliateAmount;
96 }
```

As described in the Storage Layout section in this report the GiveOrderState struct makes use of 5 storage slots. It can be reduced to 4 by applying the following code change:

Listing 4: DlnSource.sol (Lines 94,95)

```
86 struct GiveOrderState {
87     OrderGiveStatus status;
88     // stot optimisation
89     uint160 giveTokenAddress;
90     uint88 nativeFixFee;
91     uint48 takeChainId;
92     uint208 percentFee;
93     address affiliateBeneficiary;
94     uint128 giveAmount;
95     uint128 affiliateAmount;
96 }
```

This still leaves enough space for the giveAmount and affiliateAmount variables, as 340282366920938463463e18 is a value big enough and the gas cost of createOrder() calls is reduced:

BEFORE

Solc version: 0.8.17	Optimizer enabled: true	Runs: 9999	Block limit: 30000000 gas			
Methods						
Contract	Method	Min	Max	Avg	# calls	usd (avg)
DlnDestination	fulfillOrder	120823	155231	138113	36	-
DlnDestination	patchOrderTake	76430	76874	76694	5	-
DlnDestination	sendBatchEvmUnlock	-	-	263811	2	-
DlnDestination	sendEvmOrderCancel	197936	400868	227117	7	-
DlnDestination	sendEvmUnlock	-	-	153371	7	-
DlnDestination	setDlnSourceAddress	42653	79653	61153	4	-
DlnSource	createOrder	187595	249062	209755	60	-
DlnSource	patchOrderGive	66319	110464	88392	8	-
DlnSource	setDlnDestinationAddress	42654	79654	61154	4	-
DlnSource	updateGlobalFee	-	-	39663	3	-
DlnSource	withdrawFee	41933	46633	44283	4	-

AFTER THE SUGGESTED CHANGE

Solc version: 0.8.17	Optimizer enabled: true	Runs: 9999	Block limit: 30000000 gas			
Methods						
Contract	Method	Min	Max	Avg	# calls	usd (avg)
DlnDestination	fulfillOrder	120823	155231	138113	36	-
DlnDestination	patchOrderTake	76430	76874	76694	5	-
DlnDestination	sendBatchEvmUnlock	-	-	263811	2	-
DlnDestination	sendEvmOrderCancel	197936	400868	227117	7	-
DlnDestination	sendEvmUnlock	-	-	153371	7	-
DlnDestination	setDlnSourceAddress	42653	79653	61153	4	-
DlnSource	createOrder	165955	227422	188115	60	-
DlnSource	patchOrderGive	66319	110464	88392	8	-
DlnSource	setDlnDestinationAddress	42654	79654	61154	4	-
DlnSource	updateGlobalFee	-	-	39663	3	-
DlnSource	withdrawFee	41933	46633	44283	4	-

Risk Level:

Likelihood - 1

Impact - 1

Recommendation:

It is recommended to pack `giveAmount` and `affiliateAmount` within the same storage slot by declaring them as `uint128` in the `GiveOrderState` struct.

Remediation Plan:

ACKNOWLEDGED: The DeBridge team acknowledged this recommendation because they do not believe that `uint128 giveAmount` is enough. Order creation occurs without setting an affiliate reward in most cases, and an order without affiliate uses only 3 slots.

3.3 (HAL-03) UNUSED DECLARED CUSTOM ERRORS - INFORMATIONAL

Description:

The following custom errors are declared in the smart contracts in scope and not referenced anywhere else in the project:

Listing 5: DlnBase.sol

```
102 error WrongAutoArgument();
```

Listing 6: DlnBase.sol

```
110 error TheSameFromTo();
```

Listing 7: DlnDestination.sol

```
 70 error WrongToken();
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to revisit all the custom errors declared and remove those that are not planned to be used.

Remediation Plan:

SOLVED: The DeBridge team solved the issue in the following PR.

3.4 (HAL-04) UNUSED CONSTANT STATE VARIABLE - INFORMATIONAL

Description:

The following constant state variable is declared in the DlnBase smart contract and not referenced anywhere else in the project hierarchy:

Listing 8: DlnBase.sol

```
37 uint256 public constant SOLANA_CHAIN_ID = 7565164;
```

Recommendation:

It is recommended to remove the SOLANA_CHAIN_ID state variable from the DlnBase smart contract in order to reduce the deployment gas costs.

Remediation Plan:

SOLVED: The DeBridge team solved the issue in the following PR.

3.5 (HAL-05) UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0 - INFORMATIONAL

Description:

As `i` is an `uint256`, it is already initialized to 0. `uint256 i = 0` reassigns the 0 to `i` which wastes gas.

Code Location:

DlnDestination.sol
- Line 202:
```
for (uint256 i = 0; i < length; ++i){
```
- Line 223:
```
for (uint256 i = 0; i < length; ++i){
```

DlnSource.sol
- Line 256:
```
for (uint256 i = 0; i < length; ++i){
```
- Line 287:
```
for (uint256 i = 0; i < length; ++i){
```
- Line 368:
```
for (uint256 i = 0; i < length; ++i){
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended not to initialize `uint` variables to 0 to reduce the gas costs. For example, use instead:
```
for (uint256 i; i < length; ++i){
```

Remediation Plan:

SOLVED: The DeBridge team solved the issue in the following PR.

MANUAL TESTING

In the manual testing phase, the following scenarios were tested:

1. As the smart contracts make use of `call()` to perform native assets transfers, every function that uses `call()` (`_safeTransferETH()` and `_safeTransferEthOrToken()`) were reviewed in depth in order to make sure that there is no possibility for a re-entrancy attack.
2. Order creation (`createOrder()`) flow was deeply reviewed. The `DlnSource` smart contract prevents users to create 2 orders with the same `orderId` thanks to the `masterNonce[_sender]` which is increased every time an order is created.
3. `OrderTakeState.status` was fully reviewed making sure it's updated correctly at the right time, for example, preventing some malicious user to call `sendBatchEvmUnlock()` with the same orderId repeated in the `bytes32[] memory _orderIds` array.
4. Access controls were checked in every function. For example:
    a) `sendEvmUnlock()` can only be called by the `taker`.
    b) `sendBatchEvmUnlock()` can only be called by the `taker`.
    c) `sendSolanaUnlock()` can only be called by the `taker`.
    d) `sendEvmOrderCancel()` can only be called by the `orderAuthorityAddressDst`.
    e) `sendSolanaOrderCancel()` can only be called by the `orderAuthorityAddressDst`.
    f) `patchOrderTake()` can only be called by the `orderAuthorityAddressDst`.
    g) `createOrder()` can be called by anyone willing to create an order.
    h) `claimBatchUnlock()` can only be called by `DlnDestinationAddress` from the take chain.
    i) `claimUnlock()` can only be called by `DlnDestinationAddress` from the take chain.
    j) `claimBatchCancel()` can only be called by `DlnDestinationAddress` from the take chain.
    k) `claimCancel()` can only be called by `DlnDestinationAddress` from the take chain.
    l) `patchOrderGive()` can only be called by `givePatchAuthoritySrc`.
5. Permits used (`IERC20Permit`) were reviewed in order to make sure that there is no room for signature replay attacks.

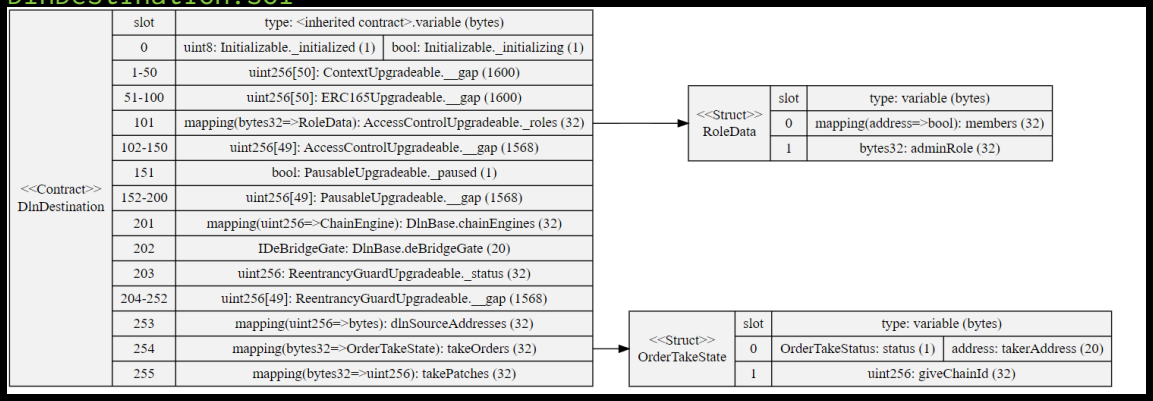
6. `abi.encodePacked()` usage across the code-base was deeply reviewed making sure that no dynamic parameters are used consecutively, preventing any possibility of hash collisions.
7. Event emissions were reviewed.
8. `msg.value` usage was also deeply reviewed, especially in the `_sendCrossChainMessage()` function.
9. Fees are correctly handled in all the required functions.

MANUAL TESTING

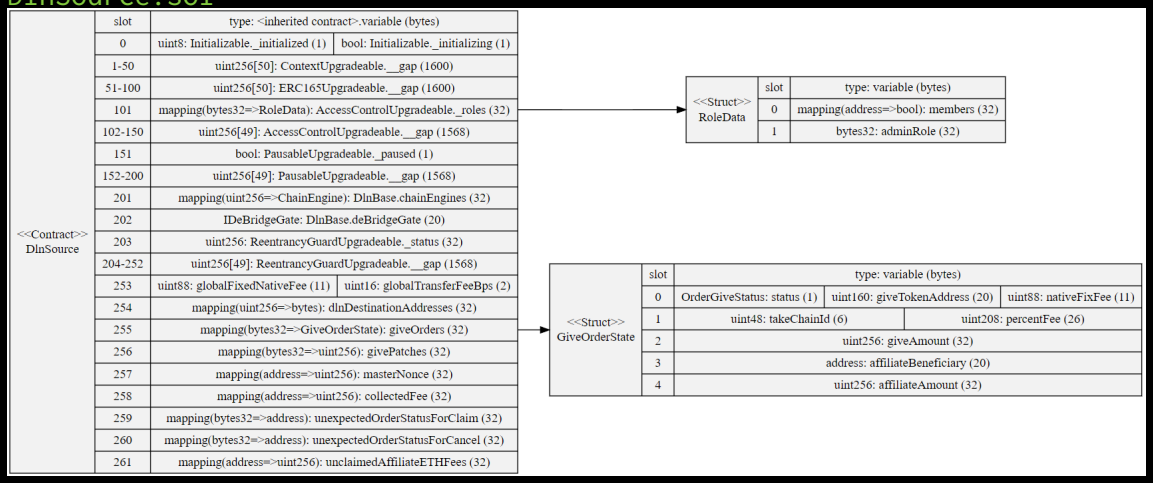
STORAGE LAYOUT

Halborn analyzed storage layout of the smart contracts in scope with tools like sol2uml to check for gas optimizations:

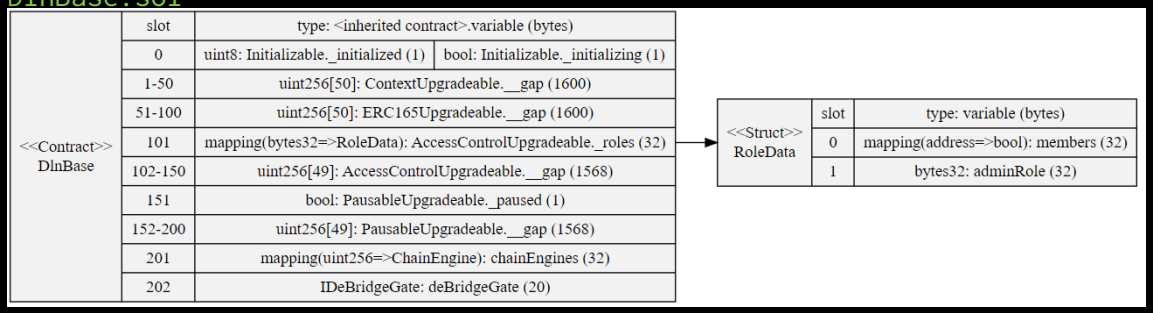
DlnDestination.sol



DlnSource.sol



DlnBase.sol



In order to further improve gas efficiency, it is recommended to apply the following change in the GiveOrderState struct:

CURRENT CODE

Listing 9: DlnSource.sol (Lines 92,94)

```
86 struct GiveOrderState {
87     OrderGiveStatus status;
88     // stot optimisation
89     uint160 giveTokenAddress;
90     uint88 nativeFixFee;
91     uint48 takeChainId;
92     uint208 percentFee;
93     uint256 giveAmount;
94     address affiliateBeneficiary;
95     uint256 affiliateAmount;
96 }
```

SUGGESTED CODE

Listing 10: DlnSource.sol (Lines 94,95)

```
86 struct GiveOrderState {
87     OrderGiveStatus status;
88     // stot optimisation
89     uint160 giveTokenAddress;
90     uint88 nativeFixFee;
91     uint48 takeChainId;
92     uint208 percentFee;
93     address affiliateBeneficiary;
94     uint128 giveAmount;
95     uint128 affiliateAmount;
96 }
```

STORAGE LAYOUT

BEFORE

Solc version: 0.8.17		Optimizer enabled: true		Runs: 9999	Block limit: 30000000 gas	
Methods						
Contract	**Method**	**Min**	**Max**	**Avg**	**# calls**	**usd (avg)**
DlnDestination	fulfillOrder	120823	155231	138113	36	-
DlnDestination	patchOrderTake	76430	76874	76694	5	-
DlnDestination	sendBatchEvmUnlock	-	-	263811	2	-
DlnDestination	sendEvmOrderCancel	197936	400868	227117	7	-
DlnDestination	sendEvmUnlock	-	-	153371	7	-
DlnDestination	setDlnSourceAddress	42653	79653	61153	4	-
DlnSource	createOrder	187595	249062	209755	60	-
DlnSource	patchOrderGive	66319	110464	88392	8	-
DlnSource	setDlnDestinationAddress	42654	79654	61154	4	-
DlnSource	updateGlobalFee	-	-	39663	3	-
DlnSource	withdrawFee	41933	46633	44283	4	-

AFTER THE SUGGESTED CHANGE

Solc version: 0.8.17		Optimizer enabled: true		Runs: 9999	Block limit: 30000000 gas	
Methods						
Contract	**Method**	**Min**	**Max**	**Avg**	**# calls**	**usd (avg)**
DlnDestination	fulfillOrder	120823	155231	138113	36	-
DlnDestination	patchOrderTake	76430	76874	76694	5	-
DlnDestination	sendBatchEvmUnlock	-	-	263811	2	-
DlnDestination	sendEvmOrderCancel	197936	400868	227117	7	-
DlnDestination	sendEvmUnlock	-	-	153371	7	-
DlnDestination	setDlnSourceAddress	42653	79653	61153	4	-
DlnSource	createOrder	165955	227422	188115	60	-
DlnSource	patchOrderGive	66319	110464	88392	8	-
DlnSource	setDlnDestinationAddress	42654	79654	61154	4	-
DlnSource	updateGlobalFee	-	-	39663	3	-
DlnSource	withdrawFee	41933	46633	44283	4	-

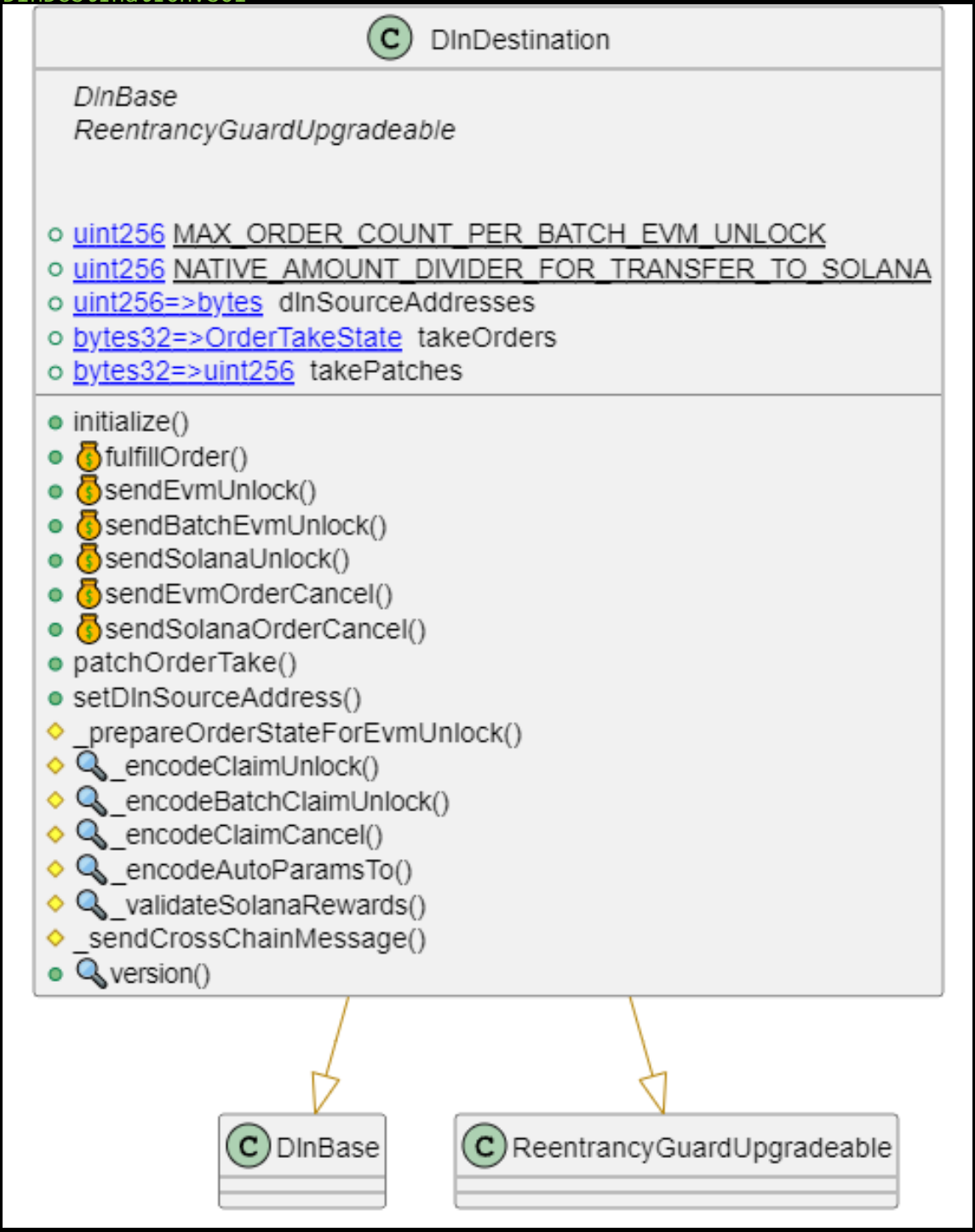
<<Struct>> GiveOrderState	slot	type: variable (bytes)		
	0	OrderGiveStatus: status (1)	uint160: giveTokenAddress (20)	uint88: nativeFixFee (11)
	1	uint48: takeChainId (6)	uint208: percentFee (26)	
	2	uint256: giveAmount (32)		
	3	address: affiliateBeneficiary (20)		
	4	uint256: affiliateAmount (32)		

<<Struct>> GiveOrderState	slot	type: variable (bytes)		
	0	OrderGiveStatus: status (1)	uint160: giveTokenAddress (20)	uint88: nativeFixFee (11)
	1	uint48: takeChainId (6)	uint208: percentFee (26)	
	2	address: affiliateBeneficiary (20)		
	3	uint128: giveAmount (16)	uint128: affiliateAmount (16)	

CONTRACT UPGRADEABILITY

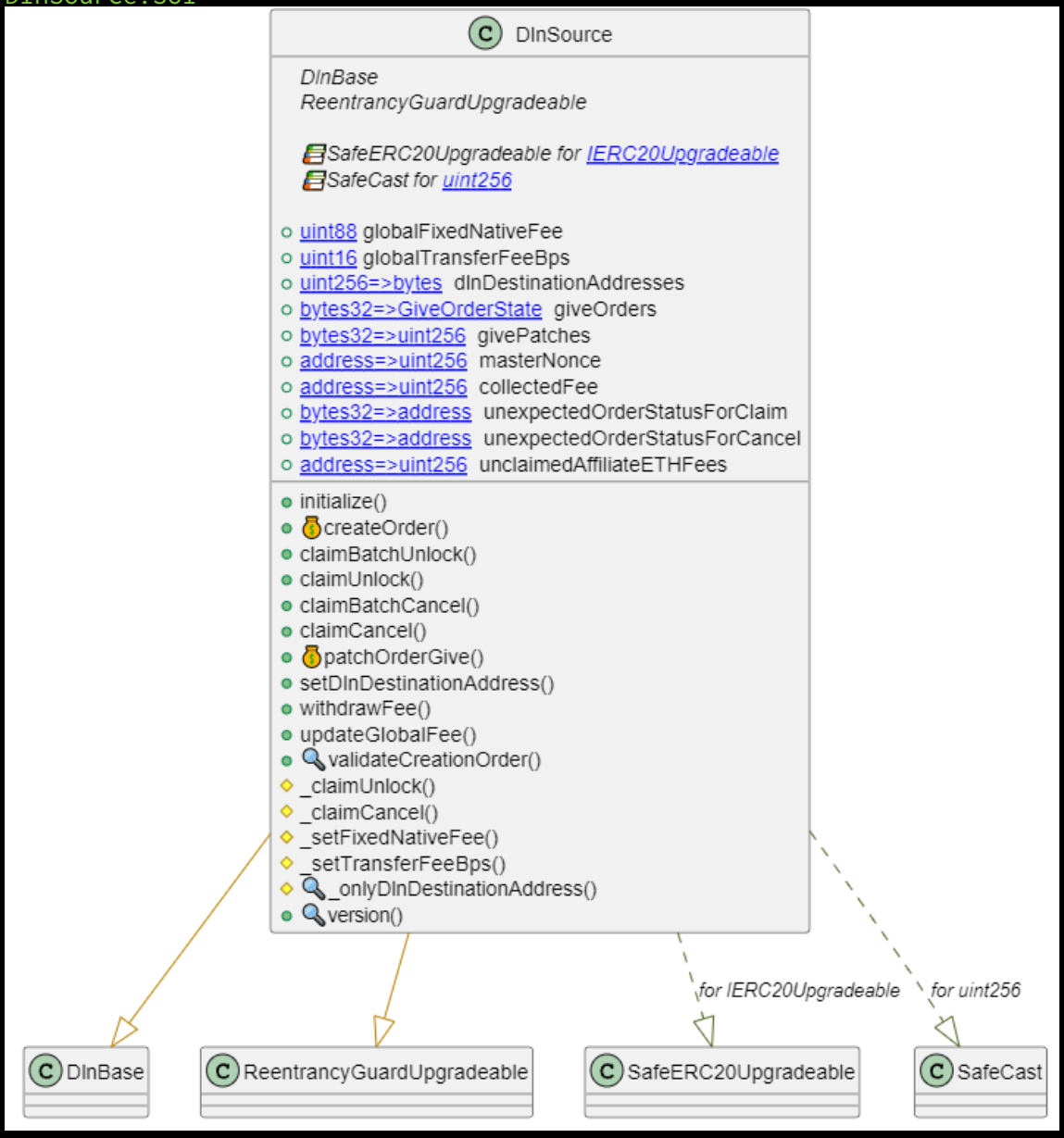
Halborn analyzed the structure of the smart contracts in scope to make sure all future upgrades are secure:

DlnDestination.sol

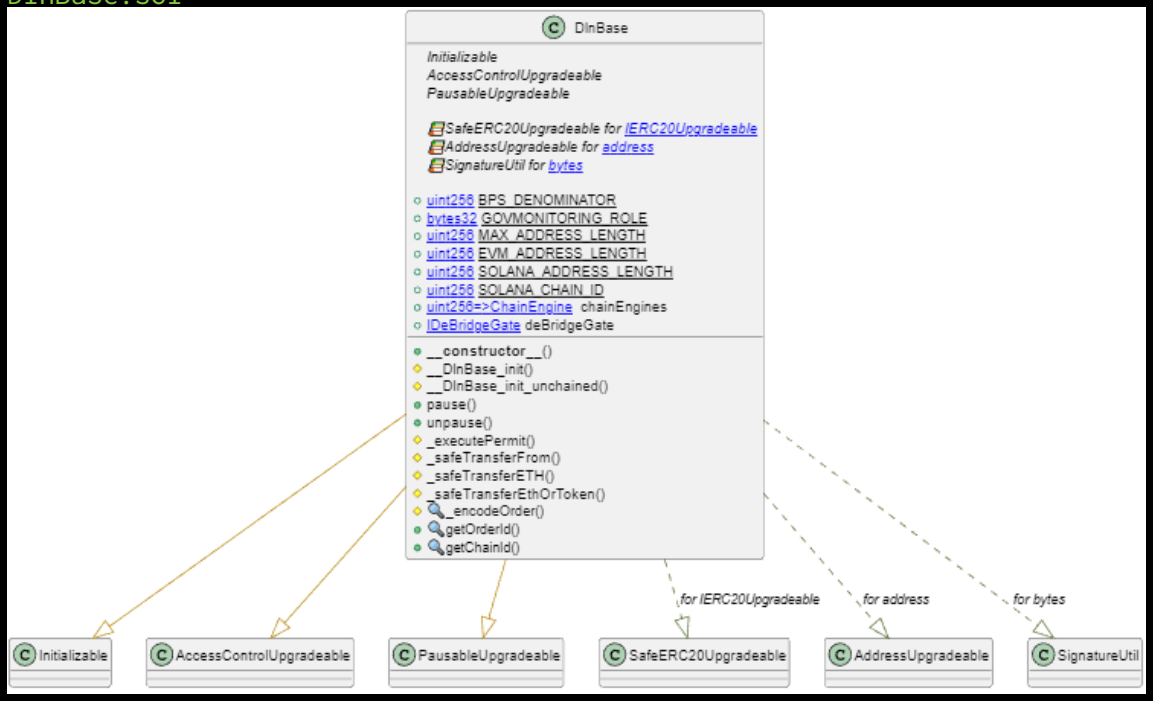


CONTRACT UPGRADEABILITY

DlnSource.sol



DlnBase.sol



1. There is no possibility of storage collisions with the proxies as DeBridge is using the standard TransparentUpgradeableProxy: https://etherscan.io/address/0xeF4fB24aD0916217251F553c0596F8Edc630EB66#code

2. The proxies and their implementations have been deployed and initialized atomically by using the OpenZeppelin Upgrades Plugins preventing any possible front-run:

```
118     dlnSourceBSC = await upgrades.deployProxy(
119       mockDlnSourceFactory,
120       [
121         deBridgeGateBSC.address,
122         deployInitParamsBSC.globalFixedNativeFee,
123         deployInitParamsBSC.globalTransferFeeBps,
124         BSC_CHAIN_ID
125       ],
126       {
127         initializer: "initializeMock",
128         kind: "transparent",
129       }
130     ) as DlnSource;
```

3. Every initialization function is correctly protected with the `initializer` modifier preventing any possible re-initializations:

Listing 11: DlnBase.sol (Lines 134,142,144)

```
134 function __DlnBase_init(IDeBridgeGate _deBridgeGate) internal
 ↳ initializer {
135     __Context_init_unchained();
136     __ERC165_init_unchained();
137     __AccessControl_init_unchained();
138     __Pausable_init_unchained();
139     __DlnBase_init_unchained(_deBridgeGate);
140 }
141
142 function __DlnBase_init_unchained(IDeBridgeGate _deBridgeGate)
143     internal
144     initializer
145 {
146     deBridgeGate = _deBridgeGate;
147     _setupRole(DEFAULT_ADMIN_ROLE, msg.sender);
148 }
```

Listing 12: DlnDestination.sol (Line 81)

```
81 function initialize(IDeBridgeGate _deBridgeGate) public
 ↳ initializer {
82     __DlnBase_init(_deBridgeGate);
83     __ReentrancyGuard_init();
84 }
```

Listing 13: DlnSource.sol (Line 160)

```
156 function initialize(
157     IDeBridgeGate _deBridgeGate,
158     uint88 _globalFixedNativeFee,
159     uint16 _globalTransferFeeBps
160 ) public initializer {
```

4. All the parent contracts are correctly initialized:

CONTRACT UPGRADEABILITY

DlnBase
- Initializable [X] (Not needed to initialize)
- AccessControlUpgradeable [X]
- PausableUpgradeable [X]

DlnDestination
- DlnBase [X]
- ReentrancyGuardUpgradeable [X]

DlnSource
- DlnBase [X]
- ReentrancyGuardUpgradeable [X]

AUTOMATED TESTING

7.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the smart contracts in scope. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified the smart contracts in the repository and was able to compile them correctly into their ABIS and binary format, Slither was run against the contracts. This tool can statically verify mathematical relationships between Solidity variables to detect invalid or inconsistent usage of the contracts' APIs across the entire code-base.

Slither results:

DlnDestination.sol

```
Reentrancy in DlnDestination.fulfillOrder(DlnBase.Order,uint256,bytes32,bytes,address) (contracts/DLN/DlnDestination.sol#101-150):
        External calls:
        - _safeTransferETH(BytesLib.toAddress(_order.receiverDst,0),takerAmount) (contracts/DLN/DlnDestination.sol#132)
              - (success) = to.call{value: value}(new bytes(0)) (contracts/DLN/DlnBase.sol#204)
        - _executePermit(takeTokenAddress,_permitEnvelope) (contracts/DLN/DlnDestination.sol#136)
              - IERC20Permit(_tokenAddress).permit(msg.sender,address(this),permitAmount,deadline,v,r,s) (contracts/DLN/DlnBase.sol#171-179)
        - _safeTransferFrom(takeTokenAddress,msg.sender,BytesLib.toAddress(_order.receiverDst,0),takerAmount) (contracts/DLN/DlnDestination.sol#137-142)
              - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
              - token.safeTransferFrom(_from,_to,_amount) (contracts/DLN/DlnBase.sol#192)
              - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
        External calls sending eth:
        - _safeTransferETH(BytesLib.toAddress(_order.receiverDst,0),takerAmount) (contracts/DLN/DlnDestination.sol#132)
              - (success) = to.call{value: value}(new bytes(0)) (contracts/DLN/DlnBase.sol#204)
        - _safeTransferFrom(takeTokenAddress,msg.sender,BytesLib.toAddress(_order.receiverDst,0),takerAmount) (contracts/DLN/DlnDestination.sol#137-142)
              - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
        State variables written after the call(s):
        - orderState.status = OrderTakeStatus.Fulfilled (contracts/DLN/DlnDestination.sol#145)
        - orderState.takerAddress = _unlockAuthority (contracts/DLN/DlnDestination.sol#146)
        - orderState.giveChainId = _order.giveChainId (contracts/DLN/DlnDestination.sol#147)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

AccessControlUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

DlnBase.deBridgeGate (contracts/DLN/DlnBase.sol#46) is never initialized. It is used in:
        - DlnSource._onlyDlnDestinationAddress() (contracts/DLN/DlnSource.sol#545-555)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse = (3 * denominator) ^ 2 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#117)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#121)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#121)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#122)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#123)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#124)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#125)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#126)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#105)
        -result = prod0 * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#132)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(_preBytes,xfalse_concatStorage_asm_0 + mload(uint256)(_postBytes + 0x20) / 0x100 ** 32 - mlength_concatStorage_asm_0 * 2) (contracts/libraries/BytesLib.sol#115-140)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#189)
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#439-509) performs a multiplication on the result of a division:
        -xfalse_equalStorage_asm_0 = xfalse_equalStorage_asm_0 / 0x100 * 0x100 (contracts/libraries/BytesLib.sol#466)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes).affiliateBeneficiary (contracts/DLN/DlnSource.sol#186) is a local variable never initialized
DlnDestination._encodeAutoParamsTo(bytes,uint256,bytes).autoParams (contracts/DLN/DlnDestination.sol#475) is a local variable never initialized
DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes).affiliateAmount (contracts/DLN/DlnSource.sol#185) is a local variable never initialized
DlnDestination.sendBatchEvmUnlock(bytes32[],address,uint256).giveChainId (contracts/DLN/DlnDestination.sol#200) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = keccak256(uint256,uint256)(0x0,0x20) + slength_concatStorage_asm_0 / 32 (contracts/libraries/BytesLib.sol#195)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: submod_concatStorage_asm_0 = 32 - slengthmod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#204)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: end_concatStorage_asm_0 = _postBytes + mlength_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#206)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (contracts/libraries/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (contracts/libraries/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (contracts/libraries/BytesLib.sol#212)
```

```
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatS
torage_asm_0 = mc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#213)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concat
torage_asm_0 < end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concat
Storage_asm_0 < end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_conca
tStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_con
catStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_conc
atStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uin
t256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(u
int256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uin
t256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uin
t256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (contracts/libraries/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uin
t256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (contracts/libraries/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatS
torage_asm_0 = sc_concatStorage_asm_0 + 1 (contracts/libraries/BytesLib.sol#215)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatS
torage_asm_0 = mc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in DlnSource._claimUnlock(bytes32,address,uint256) (contracts/DLN/DlnSource.sol#446-496):
        External calls:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        State variables written after the call(s):
        - unclaimedAffiliateETHFees[orderState.affiliateBeneficiary] += orderState.affiliateAmount (contracts/DLN/DlnSource.sol#463)
Reentrancy in DlnSource._claimUnlock(bytes32,address,uint256) (contracts/DLN/DlnSource.sol#446-496):
        External calls:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        - IERC20Upgradeable(giveTokenAddress).safeTransfer(orderState.affiliateBeneficiary,orderState.affiliateAmount) (contracts/DLN/DlnSource.sol#467-470)
        External calls sending eth:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        State variables written after the call(s):
        - collectedFee[giveTokenAddress] += orderState.percentFee (contracts/DLN/DlnSource.sol#490)
        - collectedFee[address(0)] += orderState.nativeFixFee (contracts/DLN/DlnSource.sol#491)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in DlnSource._claimUnlock(bytes32,address,uint256) (contracts/DLN/DlnSource.sol#446-496):
        External calls:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        - IERC20Upgradeable(giveTokenAddress).safeTransfer(orderState.affiliateBeneficiary,orderState.affiliateAmount) (contracts/DLN/DlnSource.sol#467-470)
        External calls sending eth:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        Event emitted after the call(s):
        - AffiliateFeePaid(_orderId,orderState.affiliateBeneficiary,orderState.affiliateAmount,giveTokenAddress) (contracts/DLN/DlnSource.sol#475-480)
        - ClaimedUnlock(_orderId,_beneficiary,amountToPay,giveTokenAddress) (contracts/DLN/DlnSource.sol#483-488)
Reentrancy in DlnDestination.fulfillOrder(DlnBase.Order,uint256,bytes32,bytes,address) (contracts/DLN/DlnDestination.sol#101-150):
        External calls:
        - _safeTransferETH(BytesLib.toAddress(_order.receiverDst,0),takerAmount) (contracts/DLN/DlnDestination.sol#132)
                - (success) = to.call{value: value}(new bytes(0)) (contracts/DLN/DlnBase.sol#204)
        - _executePermit(takeTokenAddress,_permitEnvelope) (contracts/DLN/DlnDestination.sol#136)
                - IERC20Permit(_tokenAddress).permit(msg.sender,address(this),permitAmount,deadline,v,r,s) (contracts/DLN/DlnBase.sol#171-179)
        - _safeTransferFrom(takeTokenAddress,msg.sender,BytesLib.toAddress(_order.receiverDst,0),takerAmount) (contracts/DLN/DlnDestination.sol#137-142)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - token.safeTransferFrom(_from, _to, _amount) (contracts/DLN/DlnBase.sol#192)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
        External calls sending eth:
        - _safeTransferETH(BytesLib.toAddress(_order.receiverDst,0),takerAmount) (contracts/DLN/DlnDestination.sol#132)
                - (success) = to.call{value: value}(new bytes(0)) (contracts/DLN/DlnBase.sol#204)
        - _safeTransferFrom(takeTokenAddress,msg.sender,BytesLib.toAddress(_order.receiverDst,0),takerAmount) (contracts/DLN/DlnDestination.sol#137-142)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
        Event emitted after the call(s):
        - FulfilledOrder(_order,orderId,msg.sender,_unlockAuthority) (contracts/DLN/DlnDestination.sol#149)
Reentrancy in DlnDestination.sendBatchEvmUnlock(bytes32[],address,uint256) (contracts/DLN/DlnDestination.sol#192-226):
        External calls:
        - submissionId = _sendCrossChainMessage(giveChainId,abi.encodePacked(_beneficiary),_executionFee,claimUnlockMethod) (contracts/DLN/DlnDestination.sol#216-221)
                - deBridgeGate.send{value: msg.value}(address(0),msg.value,_chainIdTo,srcAddress,,false,0,autoParams) (contracts/DLN/DlnDestination.sol#533-542)
        Event emitted after the call(s):
        - SentOrdersUnlock(_orderIds[i_scope_0],abi.encodePacked(_beneficiary),submissionId) (contracts/DLN/DlnDestination.sol#224)
Reentrancy in DlnDestination.sendEvmOrderCancel(DlnBase.Order,address,uint256) (contracts/DLN/DlnDestination.sol#292-323):
        External calls:
        - submissionId = _sendCrossChainMessage(_order.giveChainId,abi.encodePacked(_cancelBeneficiary),_executionFee,claimCancelMethod) (contracts/DLN/DlnDestination.sol#316-321)
                - deBridgeGate.send{value: msg.value}(address(0),msg.value,_chainIdTo,srcAddress,,false,0,autoParams) (contracts/DLN/DlnDestination.sol#533-542)
        Event emitted after the call(s):
        - SentOrderCancel(_order,orderId,abi.encodePacked(_cancelBeneficiary),submissionId) (contracts/DLN/DlnDestination.sol#322)
Reentrancy in DlnDestination.sendEvmUnlock(bytes32,address,uint256) (contracts/DLN/DlnDestination.sol#162-179):
        External calls:
        - submissionId = _sendCrossChainMessage(giveChainId,abi.encodePacked(_beneficiary),_executionFee,claimUnlockMethod) (contracts/DLN/DlnDestination.sol#171-176)
                - deBridgeGate.send{value: msg.value}(address(0),msg.value,_chainIdTo,srcAddress,,false,0,autoParams) (contracts/DLN/DlnDestination.sol#533-542)
        Event emitted after the call(s):
        - SentOrdersUnlock(_orderId,abi.encodePacked(_beneficiary),submissionId) (contracts/DLN/DlnDestination.sol#178)
Reentrancy in DlnDestination.sendSolanaOrderCancel(DlnBase.Order,bytes32,uint256,uint64,uint64) (contracts/DLN/DlnDestination.sol#337-384):
        External calls:
        - submissionId = _sendCrossChainMessage(_order.giveChainId,abi.encodePacked(_cancelBeneficiary),_executionFee,claimCancelMethod) (contracts/DLN/DlnDestination.sol#376-381)
                - deBridgeGate.send{value: msg.value}(address(0),msg.value,_chainIdTo,srcAddress,,false,0,autoParams) (contracts/DLN/DlnDestination.sol#533-542)
        Event emitted after the call(s):
        - SentOrderCancel(_order,orderId,abi.encodePacked(_cancelBeneficiary),submissionId) (contracts/DLN/DlnDestination.sol#383)
Reentrancy in DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64) (contracts/DLN/DlnDestination.sol#239-277):
        External calls:
        - submissionId = _sendCrossChainMessage(orderState.giveChainId,abi.encodePacked(_beneficiary),_executionFee,claimUnlockMethod) (contracts/DLN/DlnDestination.sol#269-274)
                - deBridgeGate.send{value: msg.value}(address(0),msg.value,_chainIdTo,srcAddress,,false,0,autoParams) (contracts/DLN/DlnDestination.sol#533-542)
        Event emitted after the call(s):
        - SentOrdersUnlock(orderId,abi.encodePacked(_beneficiary),submissionId) (contracts/DLN/DlnDestination.sol#276)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

SignatureUtil.parseSignature(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#32-49) uses assembly
        - INLINE ASM (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#41-45)
SignatureUtil.toUint256(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51-61) uses assembly
        - INLINE ASM (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#58-60)
AddressUpgradeable._revert(bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#206-218) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#211-214)
StringsUpgradeable.toString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#18-38) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#24-26)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#30-32)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#66-70)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#86-93)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#100-109)
DlnBase.getChainId() (contracts/DLN/DlnBase.sol#336-340) uses assembly
        - INLINE ASM (contracts/DLN/DlnBase.sol#337-339)
BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#13-89) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#23-86)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#92-225)
BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#228-295) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#242-292)
BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#297-306) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#301-303)
BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#308-317) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#312-314)
BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#319-328) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#323-325)
BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#330-339) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#334-336)
BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#341-350) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#345-347)
BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#352-361) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#356-358)
BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#363-372) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#367-369)
BytesLib.toUint256(bytes,uint256) (contracts/libraries/BytesLib.sol#374-383) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#378-380)
BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#385-394) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#389-391)
BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#396-437) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#399-434)
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#439-509) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#449-506)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.8.0<0.9.0', '^0.8.0', '^0.8.1', '^0.8.17', '^0.8.2', '^0.8.7']
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/interfaces/ICallProxy.sol#2)
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/interfaces/IDeBridgeGate.sol#2)
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/libraries/Flags.sol#2)
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#2)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#4)
        - ^0.8.17 (contracts/DLN/DlnBase.sol#2)
        - ^0.8.17 (contracts/DLN/DlnDestination.sol#2)
        - ^0.8.17 (contracts/DLN/DlnSource.sol#2)
        - ^0.8.7 (contracts/interfaces/IERC20Permit.sol#3)
        - >=0.8.0<0.9.0 (contracts/libraries/BytesLib.sol#9)
        - ^0.8.7 (contracts/libraries/EncodeSolanaDlnMessage.sol#2)
        - ^0.8.0 (contracts/libraries/SafeCast.sol#5)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
```

AUTOMATED TESTING

```
AccessControlUpgradeable._AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#145-147) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#155-162) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#194-204) is never used and should be removed
BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#13-89) is never used and should be removed
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) is never used and should be removed
BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#396-437) is never used and should be removed
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#439-509) is never used and should be removed
BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#228-295) is never used and should be removed
BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#363-372) is never used and should be removed
BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#319-328) is never used and should be removed
BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#330-339) is never used and should be removed
BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#341-350) is never used and should be removed
BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#308-317) is never used and should be removed
BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#352-361) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
Flags.getFlag(uint256,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/Flags.sol#24-30) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#144-150) is never used and should be removed
Initializable._getInitializedVersion() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#155-157) is never used and should be removed
Initializable._isInitializing() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#162-164) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#34-37) is never used and should be removed
MathUpgradeable.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#45-49) is never used and should be removed
MathUpgradeable.log10(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#258-290) is never used and should be removed
MathUpgradeable.log10(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#296-301) is never used and should be removed
MathUpgradeable.log2(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#240-241) is never used and should be removed
MathUpgradeable.log2(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#247-252) is never used and should be removed
MathUpgradeable.log256(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#309-333) is never used and should be removed
MathUpgradeable.log256(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#338-344) is never used and should be removed
MathUpgradeable.max(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#19-21) is never used and should be removed
MathUpgradeable.min(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#26-28) is never used and should be removed
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) is never used and should be removed
MathUpgradeable.mulDiv(uint256,uint256,uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#140-151) is never used and should be removed
MathUpgradeable.sqrt(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#158-185) is never used and should be removed
MathUpgradeable.sqrt(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#194-199) is never used and should be removed
PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is never used and should be removed
SafeCast.toInt104(int256) (contracts/libraries/SafeCast.sol#901-904) is never used and should be removed
SafeCast.toInt112(int256) (contracts/libraries/SafeCast.sol#883-886) is never used and should be removed
SafeCast.toInt120(int256) (contracts/libraries/SafeCast.sol#865-868) is never used and should be removed
SafeCast.toInt128(int256) (contracts/libraries/SafeCast.sol#847-850) is never used and should be removed
SafeCast.toInt136(int256) (contracts/libraries/SafeCast.sol#829-832) is never used and should be removed
SafeCast.toInt144(int256) (contracts/libraries/SafeCast.sol#811-814) is never used and should be removed
SafeCast.toInt152(int256) (contracts/libraries/SafeCast.sol#793-796) is never used and should be removed
SafeCast.toInt16(int256) (contracts/libraries/SafeCast.sol#1099-1102) is never used and should be removed
SafeCast.toInt160(int256) (contracts/libraries/SafeCast.sol#775-778) is never used and should be removed
SafeCast.toInt168(int256) (contracts/libraries/SafeCast.sol#757-760) is never used and should be removed
SafeCast.toInt176(int256) (contracts/libraries/SafeCast.sol#739-742) is never used and should be removed
SafeCast.toInt184(int256) (contracts/libraries/SafeCast.sol#721-724) is never used and should be removed
SafeCast.toInt192(int256) (contracts/libraries/SafeCast.sol#703-706) is never used and should be removed
SafeCast.toInt200(int256) (contracts/libraries/SafeCast.sol#685-688) is never used and should be removed
SafeCast.toInt208(int256) (contracts/libraries/SafeCast.sol#667-670) is never used and should be removed
SafeCast.toInt216(int256) (contracts/libraries/SafeCast.sol#649-652) is never used and should be removed
SafeCast.toInt224(int256) (contracts/libraries/SafeCast.sol#631-634) is never used and should be removed
SafeCast.toInt232(int256) (contracts/libraries/SafeCast.sol#613-616) is never used and should be removed
SafeCast.toInt24(int256) (contracts/libraries/SafeCast.sol#1081-1084) is never used and should be removed
SafeCast.toInt240(int256) (contracts/libraries/SafeCast.sol#595-598) is never used and should be removed
SafeCast.toInt248(int256) (contracts/libraries/SafeCast.sol#577-580) is never used and should be removed
SafeCast.toInt256(int256) (contracts/libraries/SafeCast.sol#1131-1135) is never used and should be removed
SafeCast.toInt32(int256) (contracts/libraries/SafeCast.sol#1063-1066) is never used and should be removed
SafeCast.toInt40(int256) (contracts/libraries/SafeCast.sol#1045-1048) is never used and should be removed
SafeCast.toInt48(int256) (contracts/libraries/SafeCast.sol#1027-1030) is never used and should be removed
SafeCast.toInt56(int256) (contracts/libraries/SafeCast.sol#1009-1012) is never used and should be removed
SafeCast.toInt64(int256) (contracts/libraries/SafeCast.sol#991-994) is never used and should be removed
SafeCast.toInt72(int256) (contracts/libraries/SafeCast.sol#973-976) is never used and should be removed
SafeCast.toInt8(int256) (contracts/libraries/SafeCast.sol#1117-1120) is never used and should be removed
SafeCast.toInt80(int256) (contracts/libraries/SafeCast.sol#955-958) is never used and should be removed
SafeCast.toInt88(int256) (contracts/libraries/SafeCast.sol#937-940) is never used and should be removed
SafeCast.toInt96(int256) (contracts/libraries/SafeCast.sol#919-922) is never used and should be removed
SafeCast.toUint104(uint256) (contracts/libraries/SafeCast.sol#341-344) is never used and should be removed
SafeCast.toUint112(uint256) (contracts/libraries/SafeCast.sol#324-327) is never used and should be removed
SafeCast.toUint120(uint256) (contracts/libraries/SafeCast.sol#307-310) is never used and should be removed
SafeCast.toUint128(uint256) (contracts/libraries/SafeCast.sol#290-293) is never used and should be removed
SafeCast.toUint136(uint256) (contracts/libraries/SafeCast.sol#273-276) is never used and should be removed
SafeCast.toUint144(uint256) (contracts/libraries/SafeCast.sol#256-259) is never used and should be removed
SafeCast.toUint152(uint256) (contracts/libraries/SafeCast.sol#238-242) is never used and should be removed
SafeCast.toUint16(uint256) (contracts/libraries/SafeCast.sol#528-531) is never used and should be removed
SafeCast.toUint160(uint256) (contracts/libraries/SafeCast.sol#222-225) is never used and should be removed
SafeCast.toUint168(uint256) (contracts/libraries/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toUint176(uint256) (contracts/libraries/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toUint184(uint256) (contracts/libraries/SafeCast.sol#171-174) is never used and should be removed
SafeCast.toUint192(uint256) (contracts/libraries/SafeCast.sol#154-157) is never used and should be removed
SafeCast.toUint200(uint256) (contracts/libraries/SafeCast.sol#137-140) is never used and should be removed
SafeCast.toUint216(uint256) (contracts/libraries/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint224(uint256) (contracts/libraries/SafeCast.sol#86-89) is never used and should be removed
SafeCast.toUint232(uint256) (contracts/libraries/SafeCast.sol#69-72) is never used and should be removed
SafeCast.toUint24(uint256) (contracts/libraries/SafeCast.sol#511-514) is never used and should be removed
SafeCast.toUint240(uint256) (contracts/libraries/SafeCast.sol#52-55) is never used and should be removed
SafeCast.toUint248(uint256) (contracts/libraries/SafeCast.sol#35-38) is never used and should be removed
SafeCast.toUint256(int256) (contracts/libraries/SafeCast.sol#559-562) is never used and should be removed
SafeCast.toUint32(uint256) (contracts/libraries/SafeCast.sol#494-497) is never used and should be removed
SafeCast.toUint40(uint256) (contracts/libraries/SafeCast.sol#477-480) is never used and should be removed
SafeCast.toUint56(uint256) (contracts/libraries/SafeCast.sol#443-446) is never used and should be removed
SafeCast.toUint64(uint256) (contracts/libraries/SafeCast.sol#426-429) is never used and should be removed
SafeCast.toUint72(uint256) (contracts/libraries/SafeCast.sol#409-412) is never used and should be removed
SafeCast.toUint8(uint256) (contracts/libraries/SafeCast.sol#545-548) is never used and should be removed
SafeCast.toUint80(uint256) (contracts/libraries/SafeCast.sol#392-395) is never used and should be removed
SafeCast.toUint88(uint256) (contracts/libraries/SafeCast.sol#375-378) is never used and should be removed
SafeCast.toUint96(uint256) (contracts/libraries/SafeCast.sol#358-361) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#46-59) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#70-81) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#61-68) is never used and should be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint256,uint8,bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#83-97) is never used and should b
e removed
SignatureUtil.getUnsignedMsg(bytes32) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#13-15) is never used and should be removed
SignatureUtil.splitSignature(bytes) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#19-30) is never used and should be removed
SignatureUtil.toUint256(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#31-41) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#43-47) is never used and should be removed
StringsUpgradeable.toString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#18-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#4) allows old versions
Pragma version^0.8.17 (contracts/DLN/DlnBase.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.17 (contracts/DLN/DlnDestination.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.17 (contracts/DLN/DlnSource.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version>=0.8.0<0.9.0 (contracts/libraries/BytesLib.sol#9) is too complex
Pragma version^0.8.0 (contracts/libraries/SafeCast.sol#3) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-137):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#155-162):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#160)
Low level call in DlnBase._safeTransferETH(address,uint256) (contracts/DLN/DlnBase.sol#203-206):
        - (success) = to.call{value: value}(new bytes(0)) (contracts/DLN/DlnBase.sol#204)
Low level call in DlnSource._claimUnlock(bytes32,address,uint256) (contracts/DLN/DlnSource.sol#446-496):
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter Flags.getFlag(uint256,uint256)._packedFlags (node_modules/debridge-contracts-v1/contracts/libraries/Flags.sol#25) is not in mixedCase
Parameter Flags.getFlag(uint256,uint256)._flag (node_modules/debridge-contracts-v1/contracts/libraries/Flags.sol#26) is not in mixedCase
Parameter Flags.setFlag(uint256,uint256,bool)._packedFlags (node_modules/debridge-contracts-v1/contracts/libraries/Flags.sol#37) is not in mixedCase
Parameter Flags.setFlag(uint256,uint256,bool)._flag (node_modules/debridge-contracts-v1/contracts/libraries/Flags.sol#38) is not in mixedCase
Parameter Flags.setFlag(uint256,uint256,bool)._value (node_modules/debridge-contracts-v1/contracts/libraries/Flags.sol#39) is not in mixedCase
Parameter SignatureUtil.getUnsignedMsg(bytes32)._submissionId (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#13) is not in mixedCase
Parameter SignatureUtil.splitSignature(bytes)._signature (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#19) is not in mixedCase
Parameter SignatureUtil.parseSignature(bytes,uint256)._signatures (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#32) is not in mixedCase
Parameter SignatureUtil.toUint256(bytes,uint256)._bytes (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51) is not in mixedCase
Parameter SignatureUtil.toUint256(bytes,uint256)._offset (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#80) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable._Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
```

```
Function DlnBase.__DlnBase_init(IDeBridgeGate) (contracts/DLN/DlnBase.sol#134-140) is not in mixedCase
Parameter DlnBase.__DlnBase_init(IDeBridgeGate)._deBridgeGate (contracts/DLN/DlnBase.sol#134) is not in mixedCase
Function DlnBase.__DlnBase_init_unchained(IDeBridgeGate) (contracts/DLN/DlnBase.sol#142-148) is not in mixedCase
Parameter DlnBase.__DlnBase_init_unchained(IDeBridgeGate)._deBridgeGate (contracts/DLN/DlnBase.sol#142) is not in mixedCase
Parameter DlnBase.getOrderId(DlnBase.Order)._order (contracts/DLN/DlnBase.sol#331) is not in mixedCase
Parameter DlnDestination.initialize(IDeBridgeGate)._deBridgeGate (contracts/DLN/DlnDestination.sol#81) is not in mixedCase
Parameter DlnDestination.fulfillOrder(DlnBase.Order,uint256,bytes32,bytes,address)._order (contracts/DLN/DlnDestination.sol#102) is not in mixedCase
Parameter DlnDestination.fulfillOrder(DlnBase.Order,uint256,bytes32,bytes,address)._fulFillAmount (contracts/DLN/DlnDestination.sol#103) is not in mixedCase
Parameter DlnDestination.fulfillOrder(DlnBase.Order,uint256,bytes32,bytes,address)._orderId (contracts/DLN/DlnDestination.sol#104) is not in mixedCase
Parameter DlnDestination.fulfillOrder(DlnBase.Order,uint256,bytes32,bytes,address)._permitEnvelope (contracts/DLN/DlnDestination.sol#105) is not in mixedCase
Parameter DlnDestination.fulfillOrder(DlnBase.Order,uint256,bytes32,bytes,address)._unlockAuthority (contracts/DLN/DlnDestination.sol#106) is not in mixedCase
Parameter DlnDestination.sendEvmUnlock(bytes32,address,uint256)._orderId (contracts/DLN/DlnDestination.sol#163) is not in mixedCase
Parameter DlnDestination.sendEvmUnlock(bytes32,address,uint256)._beneficiary (contracts/DLN/DlnDestination.sol#164) is not in mixedCase
Parameter DlnDestination.sendEvmUnlock(bytes32,address,uint256)._executionFee (contracts/DLN/DlnDestination.sol#165) is not in mixedCase
Parameter DlnDestination.sendBatchEvmUnlock(bytes32[],address,uint256)._orderIds (contracts/DLN/DlnDestination.sol#193) is not in mixedCase
Parameter DlnDestination.sendBatchEvmUnlock(bytes32[],address,uint256)._beneficiary (contracts/DLN/DlnDestination.sol#194) is not in mixedCase
Parameter DlnDestination.sendBatchEvmUnlock(bytes32[],address,uint256)._executionFee (contracts/DLN/DlnDestination.sol#195) is not in mixedCase
Parameter DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64)._order (contracts/DLN/DlnDestination.sol#240) is not in mixedCase
Parameter DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64)._beneficiary (contracts/DLN/DlnDestination.sol#241) is not in mixedCase
Parameter DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64)._executionFee (contracts/DLN/DlnDestination.sol#242) is not in mixedCase
Parameter DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64)._solanaExternalCallReward1 (contracts/DLN/DlnDestination.sol#243) is not in mixedCase
Parameter DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64)._solanaExternalCallReward2 (contracts/DLN/DlnDestination.sol#244) is not in mixedCase
Parameter DlnDestination.sendEvmOrderCancel(DlnBase.Order,address,uint256)._order (contracts/DLN/DlnDestination.sol#293) is not in mixedCase
Parameter DlnDestination.sendEvmOrderCancel(DlnBase.Order,address,uint256)._cancelBeneficiary (contracts/DLN/DlnDestination.sol#294) is not in mixedCase
Parameter DlnDestination.sendEvmOrderCancel(DlnBase.Order,address,uint256)._executionFee (contracts/DLN/DlnDestination.sol#295) is not in mixedCase
Parameter DlnDestination.sendSolanaOrderCancel(DlnBase.Order,bytes32,uint256,uint64,uint64)._order (contracts/DLN/DlnDestination.sol#338) is not in mixedCase
Parameter DlnDestination.sendSolanaOrderCancel(DlnBase.Order,bytes32,uint256,uint64,uint64)._cancelBeneficiary (contracts/DLN/DlnDestination.sol#339) is not in mixedCase
Parameter DlnDestination.sendSolanaOrderCancel(DlnBase.Order,bytes32,uint256,uint64,uint64)._executionFee (contracts/DLN/DlnDestination.sol#340) is not in mixedCase
Parameter DlnDestination.sendSolanaOrderCancel(DlnBase.Order,bytes32,uint256,uint64,uint64)._reward1 (contracts/DLN/DlnDestination.sol#341) is not in mixedCase
Parameter DlnDestination.sendSolanaOrderCancel(DlnBase.Order,bytes32,uint256,uint64,uint64)._reward2 (contracts/DLN/DlnDestination.sol#342) is not in mixedCase
Parameter DlnDestination.patchOrderTake(DlnBase.Order,uint256)._order (contracts/DLN/DlnDestination.sol#395) is not in mixedCase
Parameter DlnDestination.patchOrderTake(DlnBase.Order,uint256)._newSubtrahend (contracts/DLN/DlnDestination.sol#395) is not in mixedCase
Parameter DlnDestination.setDlnSourceAddress(uint256,bytes,DlnBase.ChainEngine)._chainIdFrom (contracts/DLN/DlnDestination.sol#415) is not in mixedCase
Parameter DlnDestination.setDlnSourceAddress(uint256,bytes,DlnBase.ChainEngine)._dlnSourceAddress (contracts/DLN/DlnDestination.sol#415) is not in mixedCase
Parameter DlnDestination.setDlnSourceAddress(uint256,bytes,DlnBase.ChainEngine)._chainEngine (contracts/DLN/DlnDestination.sol#415) is not in mixedCase
Parameter DlnSource.initialize(IDeBridgeGate,uint88,uint16)._deBridgeGate (contracts/DLN/DlnSource.sol#157) is not in mixedCase
Parameter DlnSource.initialize(IDeBridgeGate,uint88,uint16)._globalFixedNativeFee (contracts/DLN/DlnSource.sol#158) is not in mixedCase
Parameter DlnSource.initialize(IDeBridgeGate,uint88,uint16)._globalTransferFeeBps (contracts/DLN/DlnSource.sol#159) is not in mixedCase
Parameter DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes)._orderCreation (contracts/DLN/DlnSource.sol#176) is not in mixedCase
Parameter DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes)._affiliateFee (contracts/DLN/DlnSource.sol#177) is not in mixedCase
Parameter DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes)._referralCode (contracts/DLN/DlnSource.sol#178) is not in mixedCase
Parameter DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes)._permitEnvelope (contracts/DLN/DlnSource.sol#179) is not in mixedCase
Parameter DlnSource.claimBatchUnlock(bytes32[],address)._orderIds (contracts/DLN/DlnSource.sol#249) is not in mixedCase
Parameter DlnSource.claimBatchUnlock(bytes32[],address)._beneficiary (contracts/DLN/DlnSource.sol#249) is not in mixedCase
Parameter DlnSource.claimUnlock(bytes32,address)._orderId (contracts/DLN/DlnSource.sol#266) is not in mixedCase
Parameter DlnSource.claimUnlock(bytes32,address)._beneficiary (contracts/DLN/DlnSource.sol#266) is not in mixedCase
Parameter DlnSource.claimBatchCancel(bytes32[],address)._orderIds (contracts/DLN/DlnSource.sol#280) is not in mixedCase
Parameter DlnSource.claimBatchCancel(bytes32[],address)._beneficiary (contracts/DLN/DlnSource.sol#280) is not in mixedCase
Parameter DlnSource.claimCancel(bytes32,address)._orderId (contracts/DLN/DlnSource.sol#297) is not in mixedCase
Parameter DlnSource.claimCancel(bytes32,address)._beneficiary (contracts/DLN/DlnSource.sol#297) is not in mixedCase
Parameter DlnSource.patchOrderGive(DlnBase.Order,uint256,bytes)._order (contracts/DLN/DlnSource.sol#316) is not in mixedCase
Parameter DlnSource.patchOrderGive(DlnBase.Order,uint256,bytes)._addGiveAmount (contracts/DLN/DlnSource.sol#317) is not in mixedCase
Parameter DlnSource.patchOrderGive(DlnBase.Order,uint256,bytes)._permitEnvelope (contracts/DLN/DlnSource.sol#318) is not in mixedCase
Parameter DlnSource.setDlnDestinationAddress(uint256,bytes,DlnBase.ChainEngine)._chainIdTo (contracts/DLN/DlnSource.sol#353) is not in mixedCase
Parameter DlnSource.setDlnDestinationAddress(uint256,bytes,DlnBase.ChainEngine)._dlnDestinationAddress (contracts/DLN/DlnSource.sol#353) is not in mixedCase
Parameter DlnSource.setDlnDestinationAddress(uint256,bytes,DlnBase.ChainEngine)._chainEngine (contracts/DLN/DlnSource.sol#353) is not in mixedCase
Parameter DlnSource.withdrawFee(address[],address)._tokens (contracts/DLN/DlnSource.sol#366) is not in mixedCase
Parameter DlnSource.withdrawFee(address[],address)._beneficiary (contracts/DLN/DlnSource.sol#366) is not in mixedCase
Parameter DlnSource.updateGlobalFee(uint88,uint16)._globalFixedNativeFee (contracts/DLN/DlnSource.sol#381) is not in mixedCase
Parameter DlnSource.updateGlobalFee(uint88,uint16)._globalTransferFeeBps (contracts/DLN/DlnSource.sol#382) is not in mixedCase
Parameter DlnSource.validateCreationOrder(DlnSource.OrderCreation,address)._orderCreation (contracts/DLN/DlnSource.sol#393) is not in mixedCase
Parameter DlnSource.validateCreationOrder(DlnSource.OrderCreation,address)._sender (contracts/DLN/DlnSource.sol#393) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#14) is not in mixedCase

Parameter BytesLib.concat(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#15) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._bytes (contracts/libraries/BytesLib.sol#229) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._start (contracts/libraries/BytesLib.sol#230) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._length (contracts/libraries/BytesLib.sol#231) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._start (contracts/libraries/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._start (contracts/libraries/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._start (contracts/libraries/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._start (contracts/libraries/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._start (contracts/libraries/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._start (contracts/libraries/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._start (contracts/libraries/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#440) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#441) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeClaim(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._takeChainId (contracts/libraries/EncodeSolanaDlnMessage.sol#9) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeClaim(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._srcProgramId (contracts/libraries/EncodeSolanaDlnMessage.sol#10) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeClaim(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._actionBeneficiary (contracts/libraries/EncodeSolanaDlnMessage.sol#11) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeClaim(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._orderGiveTokenAddress (contracts/libraries/EncodeSolanaDlnMessage.sol#12) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeClaim(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._orderId (contracts/libraries/EncodeSolanaDlnMessage.sol#13) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeClaim(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._reward1 (contracts/libraries/EncodeSolanaDlnMessage.sol#14) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeClaim(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._reward2 (contracts/libraries/EncodeSolanaDlnMessage.sol#15) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeCancel(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._takeChainId (contracts/libraries/EncodeSolanaDlnMessage.sol#31) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeCancel(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._srcProgramId (contracts/libraries/EncodeSolanaDlnMessage.sol#32) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeCancel(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._actionBeneficiary (contracts/libraries/EncodeSolanaDlnMessage.sol#33) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeCancel(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._orderGiveTokenAddress (contracts/libraries/EncodeSolanaDlnMessage.sol#34) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeCancel(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._orderId (contracts/libraries/EncodeSolanaDlnMessage.sol#35) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeCancel(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._reward1 (contracts/libraries/EncodeSolanaDlnMessage.sol#36) is not in mixedCase
Parameter EncodeSolanaDlnMessage.encodeCancel(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64)._reward2 (contracts/libraries/EncodeSolanaDlnMessage.sol#37) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable DlnDestination._validateSolanaRewards(uint256,uint256,uint64,uint64)._solanaExternalCallReward1 (contracts/DLN/DlnDestination.sol#496) is too similar to DlnDestination._validateSolanaRewards(uint256,uint256,uint64,uint64)._solan
aExternalCallReward2 (contracts/DLN/DlnDestination.sol#497)
Variable DlnDestination._validateSolanaRewards(uint256,uint256,uint64,uint64)._solanaExternalCallReward1 (contracts/DLN/DlnDestination.sol#496) is too similar to DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64
)._solanaExternalCallReward2 (contracts/DLN/DlnDestination.sol#244)
Variable DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64)._solanaExternalCallReward1 (contracts/DLN/DlnDestination.sol#243) is too similar to DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint6
4,uint64)._solanaExternalCallReward2 (contracts/DLN/DlnDestination.sol#244)
Variable DlnDestination.sendSolanaUnlock(DlnBase.Order,bytes32,uint256,uint64,uint64)._solanaExternalCallReward1 (contracts/DLN/DlnDestination.sol#243) is too similar to DlnDestination._validateSolanaRewards(uint256,uint256,uint64,uint64
)._solanaExternalCallReward2 (contracts/DLN/DlnDestination.sol#497)
Variable DlnDestination.setDlnSourceAddress(uint256,bytes,DlnBase.ChainEngine)._chainEngine (contracts/DLN/DlnDestination.sol#415) is too similar to DlnBase.chainEngines (contracts/DLN/DlnBase.sol#42)
Variable DlnSource.setDlnDestinationAddress(uint256,bytes,DlnBase.ChainEngine)._chainEngine (contracts/DLN/DlnSource.sol#353) is too similar to DlnBase.chainEngines (contracts/DLN/DlnBase.sol#42)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#297-306) uses literals with too many digits:
        - tempAddress = mload(uint256)(_bytes + 0x20 + _start) / 0x1000000000000000000000000 (contracts/libraries/BytesLib.sol#302)
EncodeSolanaDlnMessage._encodeCall(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64,bytes8) (contracts/libraries/EncodeSolanaDlnMessage.sol#52-173) uses literals with too many digits:
        - encodedData = abi.encodePacked(reverse(_reward1),0x01f01d1f000000000000000000000101000000000000000010000000000000000010000008c97258f4e2489f1bb3d1029148e0d830b5a1399daff1084048e7bd8dbe9f859030000000000000000000000002000000000000000,_ac
tionBeneficiary,0x0000000020000000000000006ddf6e1d765a193d9cbe146ceeb79ac1cb485ed5f5b37913a8cf5857eff00a90000000020000000000000000,_orderGiveTokenAddress,0x00008c97258f4e2489f1bb3d1029148e0d830b5a1399daff1084048e7bd8dbe9f8590600000000000
0001968562fef0aab1b1d8f99d44306595cd4ba41d7cc899c007a774d23ad702ff601019f3d96f657370bf1dbb3313efba51ea7a08296ac33d77b949e1b62d538db37f20001,_actionBeneficiary,,_orderGiveTokenAddress) (contracts/libraries/EncodeSolanaDlnMessage.sol#63-83
)

EncodeSolanaDlnMessage._encodeCall(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64,bytes8) (contracts/libraries/EncodeSolanaDlnMessage.sol#52-173) uses literals with too many digits:
        - encodedData = abi.encodePacked(encodedData,0x0000000000000000000000000000000000000000000000000000000000000000006ddf6e1d765a193d9cbe146ceeb79ac1cb485ed5f5b37913a8cf5857eff00a900000100000000000000001,reverse(_reward2),,_src
ProgramId,STATE,_srcProgramId,
FEE LEDGER,_srcProgramId,FEE_LEDGER_WALLET ,_orderGiveTokenAddress,,_srcProgramId,GIVE_ORDER_STATE ,_orderId,0x00070000000000000000010000008c97258f4e2489f1bb3d1029148e0d830b5a1399daff1084048e7bd8dbe9f8590300000000000000000000000020000000000
0000,_actionBeneficiary,0x0000000020000000000000006ddf6e1d765a193d9cbe146ceeb79ac1cb485ed5f5b37913a8cf5857eff00a900000000200000000000000000) (contracts/libraries/EncodeSolanaDlnMessage.sol#86-123)
EncodeSolanaDlnMessage._encodeCall(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64,bytes8) (contracts/libraries/EncodeSolanaDlnMessage.sol#52-173) uses literals with too many digits:
        - encodedData = abi.encodePacked(encodedData,_orderGiveTokenAddress,    ,_srcProgramId,GIVE_ORDER_WALLET ,_orderId,
                                                                                                  ,_srcProgramId,AUTHORIZED_NATIVE_SENDER ,_takeChainId,,_srcProgramId,0x0d000000000000000062584959deb8a728a91cebdc187
b545d920479265052145f31fb80c73fac5aea00001968562fef0aab1b1d8f99d44306595cd4ba41d7cc899c007a774d23ad702ff60101980176896e24d940ee6f0a89d0020e1cd53aa3d17be42270bb39223f6ed75c6300018c6ecc336484fb8f32871d3c1656d832cc86eb2465048fea348cde76ae57
233100014026e8772b7640ce6fb9fd348473f43df344e3dcd89a43c93db81ee6efe08e67000106a7d517187bd16635dad40455fdc2c0c124c68f215675a5dbbacb5f080000000000107fe6a33e564217c5773c604a479581564c5e4c12465d65c9374ee2190f5ee400019f3d96f657370bf1dbb3313ef
ba51ea7a08296ac33d77b949e1b62d538db37f20001,_actionBeneficiary,0x00011507b8f891ebbfc57577d4d2e6a2b52dc0a744eba2be503e686d0d07d19e6ec70001) (contracts/libraries/EncodeSolanaDlnMessage.sol#126-156)
EncodeSolanaDlnMessage._encodeCall(uint256,bytes32,bytes32,bytes32,bytes32,uint64,uint64,bytes8) (contracts/libraries/EncodeSolanaDlnMessage.sol#52-173) uses literals with too many digits:
        - encodedData = abi.encodePacked(encodedData,_orderGiveTokenAddress,0x0000efe9c4afa6dc798a27b0c18e3cf0b76ad3fe8cc93764f6cb3112f9397f2cd1c6000006ddf6e1d765a193d9cbe146ceeb79ac1cb485ed5f5b37913a8cf5857eff00a900002800000000000000000,_d
iscriminator,_orderId) (contracts/libraries/EncodeSolanaDlnMessage.sol#159-169)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

DlnSource (contracts/DLN/DlnSource.sol#9-563) does not implement functions:
        - AccessControlUpgradeable.__AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52)
        - AccessControlUpgradeable.__AccessControl_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55)
        - ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19)
        - ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22)
        - DlnBase.__DlnBase_init(IDeBridgeGate) (contracts/DLN/DlnBase.sol#134-140)
        - DlnBase.__DlnBase_init_unchained(IDeBridgeGate) (contracts/DLN/DlnBase.sol#142-148)
        - ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25)
        - ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28)
        - PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36)
        - PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40)
        - ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42)
        - ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46)
        - AccessControlUpgradeable._checkRole(bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#102-104)
        - AccessControlUpgradeable._checkRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#113-126)
        - Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#144-150)
        - DlnBase._encodeOrder(DlnBase.Order) (contracts/DLN/DlnBase.sol#221-327)
        - DlnBase._executePermit(address,bytes) (contracts/DLN/DlnBase.sol#164-181)
        - Initializable._getInitializedVersion() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#155-157)
        - AccessControlUpgradeable._grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#233-238)
        - Initializable._isInitializing() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#162-164)
        - ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29)
        - ContextUpgradeable._msgSender() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#23-25)
        - ReentrancyGuardUpgradeable._nonReentrantAfter() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#69-73)
        - ReentrancyGuardUpgradeable._nonReentrantBefore() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#61-67)
        - PausableUpgradeable._pause() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#94-97)
        - PausableUpgradeable._requireNotPaused() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#76-78)
        - PausableUpgradeable._requirePaused() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#83-85)
        - AccessControlUpgradeable._revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#247-252)
        - DlnBase._safeTransferETH(address,uint256) (contracts/DLN/DlnBase.sol#203-206)
        - DlnBase._safeTransferEthOrToken(address,address,uint256) (contracts/DLN/DlnBase.sol#212-219)
        - DlnBase._safeTransferFrom(address,address,address,uint256) (contracts/DLN/DlnBase.sol#184-196)
```

```
        - AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224)
        - AccessControlUpgradeable._setupRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213)
        - PausableUpgradeable._unpause() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#106-109)
        - DlnBase.getChainId() (contracts/DLN/DlnBase.sol#336-340)
        - DlnBase.getOrderId(DlnBase.Order) (contracts/DLN/DlnBase.sol#331-333)
        - AccessControlUpgradeable.getRoleAdmin(bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#134-136)
        - AccessControlUpgradeable.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
        - AccessControlUpgradeable.hasRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#90-92)
        - DlnBase.pause() (contracts/DLN/DlnBase.sol#153-155)
        - PausableUpgradeable.paused() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#69-71)
        - AccessControlUpgradeable.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessControlUpgradeable.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
        - AccessControlUpgradeable.supportsInterface(bytes4) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#83-85)
        - DlnBase.unpause() (contracts/DLN/DlnBase.sol#158-160)
        - DlnSource.version() (contracts/DLN/DlnSource.sol#560-562)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#80) is never used in DlnDestination (contracts/DLN/DlnDestination.sol#12-551)
ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#80) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
AccessControlUpgradeable._roles (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#61) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#103) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
ReentrancyGuardUpgradeable._NOT_ENTERED (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#38) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
ReentrancyGuardUpgradeable._ENTERED (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#36) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
ReentrancyGuardUpgradeable._status (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#38) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
initialize(IDeBridgeGate) should be declared external:
        - DlnDestination.initialize(IDeBridgeGate) (contracts/DLN/DlnDestination.sol#81-84)
initialize(IDeBridgeGate,uint88,uint16) should be declared external:
        - DlnSource.initialize(IDeBridgeGate,uint88,uint16) (contracts/DLN/DlnSource.sol#156-166)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

DlnSource.sol

```
AccessControlUpgradeable._gap (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

DlnBase.deBridgeGate (contracts/DLN/DlnBase.sol#44) is never initialized. It is used in:
        - DlnSource._onlyDlnDestinationAddress() (contracts/DLN/DlnSource.sol#545-555)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse = (3 * denominator) ^ 2 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#117)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#121)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#122)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#123)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#124)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#125)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#126)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#105)
        -result = prod0 * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#132)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(_preBytes,fslot_concatStorage_asm_0 + mload(uint256)(_postBytes + 0x20) / 0x100 ** 32 - mlength_concatStorage_asm_0 * 0x100 ** 32 - newlength_concatStorage_asm_0 + mlength_concatStorage_asm_0 * 2) (contracts/libraries/BytesLib.sol#115-140)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#189)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#439-509) performs a multiplication on the result of a division:
        -fslot_equalStorage_asm_0 = fslot_equalStorage_asm_0 / 0x100 * 0x100 (contracts/libraries/BytesLib.sol#466)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes).affiliateBeneficiary (contracts/DLN/DlnSource.sol#186) is a local variable never initialized
DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes).affiliateAmount (contracts/DLN/DlnSource.sol#185) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#212)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: submod_concatStorage_asm_0 = 32 - slengthmod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#204)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + mlength_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#206)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: end_concatStorage_asm_0 = _postBytes + mlength_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#206)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (contracts/libraries/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (contracts/libraries/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (contracts/libraries/BytesLib.sol#212)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#213)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (contracts/libraries/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (contracts/libraries/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (contracts/libraries/BytesLib.sol#215)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in DlnSource._claimUnlock(bytes32,address,uint256) (contracts/DLN/DlnSource.sol#446-496):
        External calls:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        State variables written after the call(s):
        - unclaimedAffiliateETHFees[orderState.affiliateBeneficiary] += orderState.affiliateAmount (contracts/DLN/DlnSource.sol#463)
Reentrancy in DlnSource._claimUnlock(bytes32,address,uint256) (contracts/DLN/DlnSource.sol#446-496):
        External calls:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        - IERC20Upgradeable(giveTokenAddress).safeTransfer(orderState.affiliateBeneficiary,orderState.affiliateAmount) (contracts/DLN/DlnSource.sol#467-470)
        External calls sending eth:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        State variables written after the call(s):
        - collectedFee[giveTokenAddress] += orderState.percentFee (contracts/DLN/DlnSource.sol#490)
        - collectedFee[address(0)] += orderState.nativeFixFee (contracts/DLN/DlnSource.sol#491)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in DlnSource._claimUnlock(bytes32,address,uint256) (contracts/DLN/DlnSource.sol#446-496):
        External calls:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        - IERC20Upgradeable(giveTokenAddress).safeTransfer(orderState.affiliateBeneficiary,orderState.affiliateAmount) (contracts/DLN/DlnSource.sol#467-470)
        External calls sending eth:
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
        Event emitted after the call(s):
        - AffiliateFeePaid(_orderId,orderState.affiliateBeneficiary,orderState.affiliateAmount,giveTokenAddress) (contracts/DLN/DlnSource.sol#475-480)
        - ClaimedUnlock(_orderId,_beneficiary,amountToPay,giveTokenAddress) (contracts/DLN/DlnSource.sol#483-488)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
SignatureUtil.parseSignature(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#32-49) uses assembly
        - INLINE ASM (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#41-45)
SignatureUtil.toUint256(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51-61) uses assembly
        - INLINE ASM (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#58-60)
AddressUpgradeable._revert(bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#206-218) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#211-214)
StringsUpgradeable.toString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#18-38) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#24-26)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#30-32)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#66-70)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#86-93)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#100-109)
DlnBase.getChainId() (contracts/DLN/DlnBase.sol#336-340) uses assembly
        - INLINE ASM (contracts/DLN/DlnBase.sol#337-339)
BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#13-89) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#23-86)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#92-225)
BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#228-295) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#240-292)
BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#297-306) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#301-303)
BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#308-317) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#312-314)
BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#319-328) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#323-325)
BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#330-339) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#334-336)
BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#341-350) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#345-347)
BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#352-361) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#356-358)
BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#363-372) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#367-369)
BytesLib.toUint256(bytes,uint256) (contracts/libraries/BytesLib.sol#374-383) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#378-380)
BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#385-394) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#389-391)
BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#396-437) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#399-434)
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#439-509) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#449-506)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.0<0.9.0', '^0.8.0', '^0.8.1', '^0.8.17', '^0.8.2', '^0.8.7']
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/interfaces/ICallProxy.sol#2)
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/interfaces/IDeBridgeGate.sol#2)
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#2)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)

        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#4)
        - ^0.8.17 (contracts/DLN/DlnBase.sol#2)
        - ^0.8.17 (contracts/DLN/DlnSource.sol#2)
        - ^0.8.7 (contracts/interfaces/IERC20Permit.sol#3)
        - >=0.8.0<0.9.0 (contracts/libraries/BytesLib.sol#9)
        - ^0.8.0 (contracts/libraries/SafeCast.sol#5)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControlUpgradeable._AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is never used and should be removed
AccessControlUpgradeable._AccessControl_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._checkRole(bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#102-104) is never used and should be removed
AccessControlUpgradeable._checkRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#113-126) is never used and should be removed
AccessControlUpgradeable._grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#233-238) is never used and should be removed
AccessControlUpgradeable._revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#247-252) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#145-147) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#155-163) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#194-204) is never used and should be removed
BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#13-89) is never used and should be removed
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) is never used and should be removed
BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#396-437) is never used and should be removed
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#439-509) is never used and should be removed
BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#228-295) is never used and should be removed
BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#385-394) is never used and should be removed
BytesLib.toUint(bytes,uint256) (contracts/libraries/BytesLib.sol#363-372) is never used and should be removed
BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#319-328) is never used and should be removed
BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#330-339) is never used and should be removed
BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#341-350) is never used and should be removed
BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#308-317) is never used and should be removed
BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#352-361) is never used and should be removed
ContextUpgradeable._Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable._Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ContextUpgradeable._msgSender() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#23-25) is never used and should be removed
DlnBase._DlnBase_init_unchained(IDeBridgeGate) (contracts/DLN/DlnBase.sol#142-148) is never used and should be removed
DlnBase._encodeOrder(DlnBase.Order) (contracts/DLN/DlnBase.sol#221-327) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#144-150) is never used and should be removed
Initializable._getInitializedVersion() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#155-157) is never used and should be removed
Initializable._isInitializing() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#162-164) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#34-37) is never used and should be removed
MathUpgradeable.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#45-48) is never used and should be removed
MathUpgradeable.log10(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#258-290) is never used and should be removed
MathUpgradeable.log10(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#296-301) is never used and should be removed
MathUpgradeable.log2(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#205-241) is never used and should be removed
MathUpgradeable.log2(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#247-252) is never used and should be removed
MathUpgradeable.log256(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#309-333) is never used and should be removed
MathUpgradeable.log256(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#339-344) is never used and should be removed
MathUpgradeable.max(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#19-21) is never used and should be removed
MathUpgradeable.min(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#26-28) is never used and should be removed
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) is never used and should be removed
MathUpgradeable.mulDiv(uint256,uint256,uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#140-151) is never used and should be removed
MathUpgradeable.sqrt(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#158-189) is never used and should be removed
MathUpgradeable.sqrt(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#194-199) is never used and should be removed
PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is never used and should be removed
PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is never used and should be removed
PausableUpgradeable._pause() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#94-97) is never used and should be removed
PausableUpgradeable._requirePaused() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#83-85) is never used and should be removed
PausableUpgradeable._unpause() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#106-109) is never used and should be removed
ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is never used and should be removed
SafeCast.toInt104(int256) (contracts/libraries/SafeCast.sol#901-904) is never used and should be removed
SafeCast.toInt112(int256) (contracts/libraries/SafeCast.sol#883-886) is never used and should be removed
SafeCast.toInt120(int256) (contracts/libraries/SafeCast.sol#865-868) is never used and should be removed
SafeCast.toInt128(int256) (contracts/libraries/SafeCast.sol#847-850) is never used and should be removed
SafeCast.toInt136(int256) (contracts/libraries/SafeCast.sol#829-832) is never used and should be removed
SafeCast.toInt144(int256) (contracts/libraries/SafeCast.sol#811-814) is never used and should be removed
SafeCast.toInt152(int256) (contracts/libraries/SafeCast.sol#793-796) is never used and should be removed
SafeCast.toInt16(int256) (contracts/libraries/SafeCast.sol#1099-1102) is never used and should be removed
SafeCast.toInt160(int256) (contracts/libraries/SafeCast.sol#775-778) is never used and should be removed
SafeCast.toInt168(int256) (contracts/libraries/SafeCast.sol#757-760) is never used and should be removed
SafeCast.toInt176(int256) (contracts/libraries/SafeCast.sol#739-742) is never used and should be removed
SafeCast.toInt184(int256) (contracts/libraries/SafeCast.sol#721-724) is never used and should be removed
SafeCast.toInt192(int256) (contracts/libraries/SafeCast.sol#703-706) is never used and should be removed
SafeCast.toInt200(int256) (contracts/libraries/SafeCast.sol#685-688) is never used and should be removed
SafeCast.toInt208(int256) (contracts/libraries/SafeCast.sol#667-670) is never used and should be removed
SafeCast.toInt216(int256) (contracts/libraries/SafeCast.sol#649-652) is never used and should be removed
SafeCast.toInt224(int256) (contracts/libraries/SafeCast.sol#631-634) is never used and should be removed
SafeCast.toInt232(int256) (contracts/libraries/SafeCast.sol#613-616) is never used and should be removed
SafeCast.toInt24(int256) (contracts/libraries/SafeCast.sol#1081-1084) is never used and should be removed
SafeCast.toInt240(int256) (contracts/libraries/SafeCast.sol#595-598) is never used and should be removed
SafeCast.toInt248(int256) (contracts/libraries/SafeCast.sol#577-580) is never used and should be removed
SafeCast.toInt256(int256) (contracts/libraries/SafeCast.sol#131-1135) is never used and should be removed
SafeCast.toInt32(int256) (contracts/libraries/SafeCast.sol#1063-1066) is never used and should be removed
SafeCast.toInt40(int256) (contracts/libraries/SafeCast.sol#1045-1048) is never used and should be removed
SafeCast.toInt48(int256) (contracts/libraries/SafeCast.sol#1027-1030) is never used and should be removed
SafeCast.toInt56(int256) (contracts/libraries/SafeCast.sol#1009-1012) is never used and should be removed
SafeCast.toInt64(int256) (contracts/libraries/SafeCast.sol#991-994) is never used and should be removed
SafeCast.toInt72(int256) (contracts/libraries/SafeCast.sol#973-976) is never used and should be removed
SafeCast.toInt8(int256) (contracts/libraries/SafeCast.sol#1117-1120) is never used and should be removed
SafeCast.toInt80(int256) (contracts/libraries/SafeCast.sol#955-958) is never used and should be removed
SafeCast.toInt88(int256) (contracts/libraries/SafeCast.sol#937-940) is never used and should be removed
SafeCast.toInt96(int256) (contracts/libraries/SafeCast.sol#919-922) is never used and should be removed
SafeCast.toUint104(uint256) (contracts/libraries/SafeCast.sol#341-344) is never used and should be removed
SafeCast.toUint112(uint256) (contracts/libraries/SafeCast.sol#324-327) is never used and should be removed
SafeCast.toUint120(uint256) (contracts/libraries/SafeCast.sol#307-310) is never used and should be removed
SafeCast.toUint128(uint256) (contracts/libraries/SafeCast.sol#290-293) is never used and should be removed
SafeCast.toUint136(uint256) (contracts/libraries/SafeCast.sol#273-276) is never used and should be removed
SafeCast.toUint144(uint256) (contracts/libraries/SafeCast.sol#256-259) is never used and should be removed
SafeCast.toUint152(uint256) (contracts/libraries/SafeCast.sol#239-242) is never used and should be removed
SafeCast.toUint16(uint256) (contracts/libraries/SafeCast.sol#528-531) is never used and should be removed
SafeCast.toUint160(uint256) (contracts/libraries/SafeCast.sol#222-225) is never used and should be removed
SafeCast.toUint168(uint256) (contracts/libraries/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toUint176(uint256) (contracts/libraries/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toUint184(uint256) (contracts/libraries/SafeCast.sol#171-174) is never used and should be removed
SafeCast.toUint192(uint256) (contracts/libraries/SafeCast.sol#154-157) is never used and should be removed
SafeCast.toUint200(uint256) (contracts/libraries/SafeCast.sol#137-140) is never used and should be removed
SafeCast.toUint216(uint256) (contracts/libraries/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint224(uint256) (contracts/libraries/SafeCast.sol#86-89) is never used and should be removed
SafeCast.toUint232(uint256) (contracts/libraries/SafeCast.sol#69-72) is never used and should be removed
```

AUTOMATED TESTING

```
SafeCast.toUint24(uint256) (contracts/libraries/SafeCast.sol#511-514) is never used and should be removed
SafeCast.toUint240(uint256) (contracts/libraries/SafeCast.sol#52-55) is never used and should be removed
SafeCast.toUint248(uint256) (contracts/libraries/SafeCast.sol#35-38) is never used and should be removed
SafeCast.toUint256(int256) (contracts/libraries/SafeCast.sol#559-562) is never used and should be removed
SafeCast.toUint32(uint256) (contracts/libraries/SafeCast.sol#496-497) is never used and should be removed
SafeCast.toUint40(uint256) (contracts/libraries/SafeCast.sol#477-480) is never used and should be removed
SafeCast.toUint56(uint256) (contracts/libraries/SafeCast.sol#443-446) is never used and should be removed
SafeCast.toUint64(uint256) (contracts/libraries/SafeCast.sol#426-428) is never used and should be removed
SafeCast.toUint72(uint256) (contracts/libraries/SafeCast.sol#409-412) is never used and should be removed
SafeCast.toUint8(uint256) (contracts/libraries/SafeCast.sol#545-548) is never used and should be removed
SafeCast.toUint80(uint256) (contracts/libraries/SafeCast.sol#392-395) is never used and should be removed
SafeCast.toUint88(uint256) (contracts/libraries/SafeCast.sol#375-378) is never used and should be removed
SafeCast.toUint96(uint256) (contracts/libraries/SafeCast.sol#358-361) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#46-59) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#70-81) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#61-68) is never used and should be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint256,uint8,bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#83-97) is never used and should b
e removed
SafeERC20Upgradeable.safeTransferFrom(IERC20Upgradeable,address,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#30-37) is never used and should be removed
SignatureUtil.getUnsignedMsg(bytes32) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#13-15) is never used and should be removed
SignatureUtil.parseSignature(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#32-48) is never used and should be removed
SignatureUtil.splitSignature(bytes) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#19-30) is never used and should be removed
SignatureUtil.toUint256(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51-61) is never used and should be removed
StringsUpgradeable.toHexString(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#67-69) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#43-47) is never used and should be removed
StringsUpgradeable.toHexString(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#52-62) is never used and should be removed
StringsUpgradeable.toString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#18-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#4) allows old versions
Pragma version^0.8.17 (contracts/DLN/DlnBase.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.17 (contracts/DLN/DlnSource.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version>=0.8.0<0.9.0 (contracts/libraries/BytesLib.sol#9) is too complex
Pragma version^0.8.0 (contracts/libraries/SafeCast.sol#5) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-137):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#155-162):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#160)
Low level call in DlnBase._safeTransferETH(address,uint256) (contracts/DLN/DlnBase.sol#203-206):
        - (success) = to.call{value: value}(new bytes(0)) (contracts/DLN/DlnBase.sol#204)
Low level call in DlnSource._claimUnlock(bytes32,address,uint256) (contracts/DLN/DlnSource.sol#446-456):
        - (success,None) = orderState.affiliateBeneficiary.call{gas: 2300,value: orderState.affiliateAmount}(new bytes(0)) (contracts/DLN/DlnSource.sol#461)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter SignatureUtil.getUnsignedMsg(bytes32)._submissionId (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#13) is not in mixedCase
Parameter SignatureUtil.splitSignature(bytes)._signature (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#19) is not in mixedCase
Parameter SignatureUtil.parseSignature(bytes,uint256)._signatures (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#32) is not in mixedCase
Parameter SignatureUtil.toUint256(bytes,uint256)._bytes (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51) is not in mixedCase
Parameter SignatureUtil.toUint256(bytes,uint256)._offset (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable._gap (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable._gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#126) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable._gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#80) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Function DlnBase.__DlnBase_init(IDeBridgeGate) (contracts/DLN/DlnBase.sol#134-140) is not in mixedCase
Parameter DlnBase.__DlnBase_init(IDeBridgeGate)._deBridgeGate (contracts/DLN/DlnBase.sol#134) is not in mixedCase
Function DlnBase.__DlnBase_init_unchained(IDeBridgeGate) (contracts/DLN/DlnBase.sol#142-148) is not in mixedCase
Parameter DlnBase.__DlnBase_init_unchained(IDeBridgeGate)._deBridgeGate (contracts/DLN/DlnBase.sol#142) is not in mixedCase
Parameter DlnBase.getOrderId(DlnBase.Order)._order (contracts/DLN/DlnBase.sol#331) is not in mixedCase
Parameter DlnSource.initialize(IDeBridgeGate,uint8,uint16,_deBridgeGate (contracts/DLN/DlnSource.sol#157) is not in mixedCase
Parameter DlnSource.initialize(IDeBridgeGate,uint8,uint16)._globalFixedNativeFee (contracts/DLN/DlnSource.sol#158) is not in mixedCase
Parameter DlnSource.initialize(IDeBridgeGate,uint8,uint16)._globalTransferFeeBps (contracts/DLN/DlnSource.sol#159) is not in mixedCase
Parameter DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes)._orderCreation (contracts/DLN/DlnSource.sol#176) is not in mixedCase
Parameter DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes)._affiliateFee (contracts/DLN/DlnSource.sol#177) is not in mixedCase
Parameter DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes)._referralCode (contracts/DLN/DlnSource.sol#178) is not in mixedCase
Parameter DlnSource.createOrder(DlnSource.OrderCreation,bytes,uint32,bytes)._permitEnvelope (contracts/DLN/DlnSource.sol#179) is not in mixedCase
Parameter DlnSource.claimBatchUnlock(bytes32[],address)._orderIds (contracts/DLN/DlnSource.sol#249) is not in mixedCase
Parameter DlnSource.claimBatchUnlock(bytes32[],address)._beneficiary (contracts/DLN/DlnSource.sol#249) is not in mixedCase
Parameter DlnSource.claimUnlock(bytes32,address)._orderId (contracts/DLN/DlnSource.sol#266) is not in mixedCase
Parameter DlnSource.claimUnlock(bytes32,address)._beneficiary (contracts/DLN/DlnSource.sol#266) is not in mixedCase
Parameter DlnSource.claimBatchCancel(bytes32[],address)._orderIds (contracts/DLN/DlnSource.sol#280) is not in mixedCase
Parameter DlnSource.claimBatchCancel(bytes32[],address)._beneficiary (contracts/DLN/DlnSource.sol#280) is not in mixedCase
Parameter DlnSource.claimCancel(bytes32,address)._orderId (contracts/DLN/DlnSource.sol#297) is not in mixedCase
Parameter DlnSource.claimCancel(bytes32,address)._beneficiary (contracts/DLN/DlnSource.sol#297) is not in mixedCase
Parameter DlnSource.patchOrderGive(DlnBase.Order,uint256,bytes)._order (contracts/DLN/DlnSource.sol#316) is not in mixedCase
Parameter DlnSource.patchOrderGive(DlnBase.Order,uint256,bytes)._addGiveAmount (contracts/DLN/DlnSource.sol#317) is not in mixedCase
Parameter DlnSource.patchOrderGive(DlnBase.Order,uint256,bytes)._permitEnvelope (contracts/DLN/DlnSource.sol#318) is not in mixedCase
Parameter DlnSource.setDlnDestinationAddress(uint256,bytes,DlnBase.ChainEngine)._chainId (contracts/DLN/DlnSource.sol#353) is not in mixedCase
Parameter DlnSource.setDlnDestinationAddress(uint256,bytes,DlnBase.ChainEngine)._dlnDestinationAddress (contracts/DLN/DlnSource.sol#353) is not in mixedCase
Parameter DlnSource.setDlnDestinationAddress(uint256,bytes,DlnBase.ChainEngine)._chainEngine (contracts/DLN/DlnSource.sol#353) is not in mixedCase
Parameter DlnSource.withdrawFee(address[],address)._tokens (contracts/DLN/DlnSource.sol#366) is not in mixedCase
Parameter DlnSource.withdrawFee(address[],address)._beneficiary (contracts/DLN/DlnSource.sol#366) is not in mixedCase
Parameter DlnSource.updateGlobalFee(uint8,uint16)._globalFixedNativeFee (contracts/DLN/DlnSource.sol#381) is not in mixedCase
Parameter DlnSource.updateGlobalFee(uint8,uint16)._globalTransferFeeBps (contracts/DLN/DlnSource.sol#382) is not in mixedCase
Parameter DlnSource.validateCreationOrder(DlnSource.OrderCreation,address)._orderCreation (contracts/DLN/DlnSource.sol#393) is not in mixedCase
Parameter DlnSource.validateCreationOrder(DlnSource.OrderCreation,address)._sender (contracts/DLN/DlnSource.sol#393) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#14) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#15) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._bytes (contracts/libraries/BytesLib.sol#229) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._start (contracts/libraries/BytesLib.sol#230) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._length (contracts/libraries/BytesLib.sol#231) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._start (contracts/libraries/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._start (contracts/libraries/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._start (contracts/libraries/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._start (contracts/libraries/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._start (contracts/libraries/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._start (contracts/libraries/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._start (contracts/libraries/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#440) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#441) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable DlnSource.setDlnDestinationAddress(uint256,bytes,DlnBase.ChainEngine)._chainEngine (contracts/DLN/DlnSource.sol#353) is too similar to DlnBase.chainEngines (contracts/DLN/DlnBase.sol#42)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#297-306) uses literals with too many digits:
        - tempAddress = mload(uint256)(_bytes + 0x20 + _start) / 0x1000000000000000000000000 (contracts/libraries/BytesLib.sol#302)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

DlnSource (contracts/DLN/DlnSource.sol#9-563) does not implement functions:
        - AccessControlUpgradeable.__AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52)
        - AccessControlUpgradeable.__AccessControl_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55)
        - ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19)
        - ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22)
        - DlnBase.__DlnBase_init(IDeBridgeGate) (contracts/DLN/DlnBase.sol#134-140)
        - DlnBase.__DlnBase_init_unchained(IDeBridgeGate) (contracts/DLN/DlnBase.sol#142-148)
        - ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25)
        - ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28)
        - PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36)
        - PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40)
        - ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42)
        - ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46)
        - AccessControlUpgradeable._checkRole(bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#102-104)
        - AccessControlUpgradeable._checkRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#113-126)
        - Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#144-150)
        - DlnBase._encodeOrder(DlnBase.Order) (contracts/DLN/DlnBase.sol#221-327)
        - DlnBase._executePermit(address,bytes) (contracts/DLN/DlnBase.sol#164-181)
        - Initializable._getInitializedVersion() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#155-157)
        - AccessControlUpgradeable._grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#233-238)
        - Initializable._isInitializing() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#162-164)
        - ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29)
        - ContextUpgradeable._msgSender() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#23-25)
```

```
        - ReentrancyGuardUpgradeable._nonReentrantAfter() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#69-73)
        - ReentrancyGuardUpgradeable._nonReentrantBefore() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#61-67)
        - PausableUpgradeable._pause() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#94-97)
        - PausableUpgradeable._requireNotPaused() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#76-78)
        - PausableUpgradeable._requirePaused() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#83-85)
        - AccessControlUpgradeable._revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#247-252)
        - DlnBase._safeTransferETH(address,uint256) (contracts/DLN/DlnBase.sol#203-206)
        - DlnBase._safeTransferFromOrToken(address,address,uint256) (contracts/DLN/DlnBase.sol#212-219)
        - DlnBase._safeTransferFrom(address,address,address,uint256) (contracts/DLN/DlnBase.sol#184-196)
        - AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224)
        - AccessControlUpgradeable._setupRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213)
        - PausableUpgradeable._unpause() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#106-109)
        - DlnBase.getChainId() (contracts/DLN/DlnBase.sol#336-340)
        - DlnBase.getOrderId(DlnBase.Order) (contracts/DLN/DlnBase.sol#331-333)
        - AccessControlUpgradeable.getRoleAdmin(bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#134-136)
        - AccessControlUpgradeable.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
        - AccessControlUpgradeable.hasRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#90-92)
        - DlnBase.pause() (contracts/DLN/DlnBase.sol#153-155)
        - PausableUpgradeable.paused() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#69-71)
        - AccessControlUpgradeable.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessControlUpgradeable.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
        - AccessControlUpgradeable.supportsInterface(bytes4) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#83-85)
        - DlnBase.unpause() (contracts/DLN/DlnBase.sol#158-160)
        - DlnSource.version() (contracts/DLN/DlnSource.sol#560-562)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#80) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
AccessControlUpgradeable._roles (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#61) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
PausableUpgradeable._paused (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#29) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
ReentrancyGuardUpgradeable._NOT_ENTERED (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#35) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
ReentrancyGuardUpgradeable._ENTERED (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#36) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
ReentrancyGuardUpgradeable._status (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#38) is never used in DlnSource (contracts/DLN/DlnSource.sol#9-563)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
initialize(IDeBridgeGate,uint88,uint16) should be declared external:
        - DlnSource.initialize(IDeBridgeGate,uint88,uint16) (contracts/DLN/DlnSource.sol#186-166)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

DlnBase.sol

```
AccessControlUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse = (3 * denominator) ^ 2 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#117)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#121)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#122)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#123)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#124)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#125)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#126)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#105)
        -result = prod0 * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#132)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(_preBytes,fslot_concatStorage_asm_0 + mload(uint256)(_postBytes + 0x20) / 0x100 ** 32 - mlength_concatStorage_asm_0 * 0x100 ** 32 - newlength_concatStorage_asm_0 + mlength_concatStorage_asm_0 * 2) (contracts/libraries/BytesLib.sol#115-140)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(sc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#189)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#499-509) performs a multiplication on the result of a division:
        -fslot_equalStorage_asm_0 = fslot_equalStorage_asm_0 / 0x100 * 0x100 (contracts/libraries/BytesLib.sol#466)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = keccak256(uint256,uint256)(0x0,0x20) + slength_concatStorage_asm_0 / 32 (contracts/libraries/BytesLib.sol#195)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: submod_concatStorage_asm_0 = 32 - slength(mod,concatStorage_asm_0) (contracts/libraries/BytesLib.sol#204)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: end_concatStorage_asm_0 = _postBytes + slength_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#206)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (contracts/libraries/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (contracts/libraries/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (contracts/libraries/BytesLib.sol#212)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#213)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#215)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (contracts/libraries/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (contracts/libraries/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (contracts/libraries/BytesLib.sol#215)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

SignatureUtil.parseSignature(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#32-49) uses assembly
        - INLINE ASM (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#41-45)
SignatureUtil.toUint256(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51-61) uses assembly
        - INLINE ASM (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#58-60)
AddressUpgradeable._revert(bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#206-218) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#211-214)
StringsUpgradeable.toString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#18-38) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#24-26)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#30-32)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#66-70)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#86-93)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#100-109)
DlnBase.getChainId() (contracts/DLN/DlnBase.sol#336-340) uses assembly
        - INLINE ASM (contracts/DLN/DlnBase.sol#337-339)
BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#13-89) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#23-86)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#92-225)
BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#228-295) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#242-292)
BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#297-306) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#301-303)
BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#308-317) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#312-314)
BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#319-328) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#323-325)
BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#330-339) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#334-336)
BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#341-350) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#345-347)
BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#352-361) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#356-358)
BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#363-372) uses assembly
```

```
        - INLINE ASM (contracts/libraries/BytesLib.sol#367-369)
BytesLib.toUint256(bytes,uint256) (contracts/libraries/BytesLib.sol#374-383) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#378-380)
BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#385-394) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#389-391)
BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#396-437) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#399-434)
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#439-509) uses assembly
        - INLINE ASM (contracts/libraries/BytesLib.sol#449-506)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.8.0<0.9.0', '^0.8.0', '^0.8.1', '^0.8.17', '^0.8.2', '^0.8.7']
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/interfaces/IDeBridgeGate.sol#2)
        - ^0.8.7 (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#2)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#4)
        - ^0.8.17 (contracts/DLN/DlnBase.sol#2)
        - ^0.8.7 (contracts/interfaces/IERC20Permit.sol#3)
        - >=0.8.0<0.9.0 (contracts/libraries/BytesLib.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControlUpgradeable._AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is never used and should be removed
AccessControlUpgradeable._AccessControl_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable._revert(bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#206-218) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-137) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#145-147) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#155-162) is never used and should be removed
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#36-42) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#194-204) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#170-196) is never used and should be removed
BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#18-89) is never used and should be removed
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) is never used and should be removed
BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#396-437) is never used and should be removed
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#439-509) is never used and should be removed
BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#225-295) is never used and should be removed
BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#297-306) is never used and should be removed
BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#385-394) is never used and should be removed
BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#363-372) is never used and should be removed
BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#319-328) is never used and should be removed
BytesLib.toUint256(bytes,uint256) (contracts/libraries/BytesLib.sol#374-383) is never used and should be removed
BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#330-339) is never used and should be removed
BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#341-350) is never used and should be removed
BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#308-317) is never used and should be removed
BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#352-361) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed

ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
DlnBase._DlnBase_init(IDeBridgeGate) (contracts/DLN/DlnBase.sol#134-140) is never used and should be removed
DlnBase._DlnBase_init_unchained(IDeBridgeGate) (contracts/DLN/DlnBase.sol#142-148) is never used and should be removed
DlnBase._executePermit(address,bytes) (contracts/DLN/DlnBase.sol#164-181) is never used and should be removed
DlnBase._safeTransferETH(address,uint256) (contracts/DLN/DlnBase.sol#203-206) is never used and should be removed
DlnBase._safeTransferFromOrToken(address,address,address,uint256) (contracts/DLN/DlnBase.sol#212-218) is never used and should be removed
DlnBase._safeTransferFrom(address,address,address,uint256) (contracts/DLN/DlnBase.sol#184-196) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#144-150) is never used and should be removed
Initializable._getInitializedVersion() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#155-157) is never used and should be removed
Initializable._isInitializing() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#162-164) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#34-37) is never used and should be removed
MathUpgradeable.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#45-48) is never used and should be removed
MathUpgradeable.log10(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#258-290) is never used and should be removed
MathUpgradeable.log10(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#296-301) is never used and should be removed
MathUpgradeable.log2(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#205-241) is never used and should be removed
MathUpgradeable.log2(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#247-252) is never used and should be removed
MathUpgradeable.log256(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#309-333) is never used and should be removed
MathUpgradeable.log256(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#339-344) is never used and should be removed
MathUpgradeable.max(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#19-21) is never used and should be removed
MathUpgradeable.min(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#26-28) is never used and should be removed
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) is never used and should be removed
MathUpgradeable.mulDiv(uint256,uint256,uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#140-151) is never used and should be removed
MathUpgradeable.sqrt(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#158-159) is never used and should be removed
MathUpgradeable.sqrt(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#194-199) is never used and should be removed
PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is never used and should be removed
PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is never used and should be removed
SafeERC20Upgradeable._callOptionalReturn(IERC20Upgradeable,bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#105-115) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#46-59) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#70-81) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#61-68) is never used and should be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint256,uint8,bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#83-97) is never used and should b
e removed
SafeERC20Upgradeable.safeTransfer(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#22-28) is never used and should be removed
SafeERC20Upgradeable.safeTransferFrom(IERC20Upgradeable,address,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#30-37) is never used and should be removed
SignatureUtil.getUnsignedMsg(bytes32) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#13-15) is never used and should be removed
SignatureUtil.parseSignature(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#32-49) is never used and should be removed
SignatureUtil.splitSignature(bytes) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#19-30) is never used and should be removed
SignatureUtil.toUint256(bytes,uint256) (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51-61) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#43-47) is never used and should be removed
StringsUpgradeable.toString(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#18-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#4) allows old versions
Pragma version^0.8.17 (contracts/DLN/DlnBase.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version>=0.8.0<0.9.0 (contracts/libraries/BytesLib.sol#9) is too complex
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-137):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#155-162):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#160)
Low level call in DlnBase._safeTransferETH(address,uint256) (contracts/DLN/DlnBase.sol#203-206):
        - (success) = to.call{value: value}(new bytes(0)) (contracts/DLN/DlnBase.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter SignatureUtil.getUnsignedMsg(bytes32)._submissionId (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#13) is not in mixedCase
Parameter SignatureUtil.splitSignature(bytes)._signature (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#19) is not in mixedCase
Parameter SignatureUtil.parseSignature(bytes,uint256)._signatures (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#32) is not in mixedCase
Parameter SignatureUtil.toUint256(bytes,uint256)._bytes (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51) is not in mixedCase
Parameter SignatureUtil.toUint256(bytes,uint256)._offset (node_modules/debridge-contracts-v1/contracts/libraries/SignatureUtil.sol#51) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Function DlnBase._DlnBase_init(IDeBridgeGate) (contracts/DLN/DlnBase.sol#134-140) is not in mixedCase
Parameter DlnBase._DlnBase_init(IDeBridgeGate)._deBridgeGate (contracts/DLN/DlnBase.sol#134) is not in mixedCase
Function DlnBase._DlnBase_init_unchained(IDeBridgeGate) (contracts/DLN/DlnBase.sol#142-148) is not in mixedCase
Parameter DlnBase._DlnBase_init_unchained(IDeBridgeGate)._deBridgeGate (contracts/DLN/DlnBase.sol#142) is not in mixedCase
Parameter DlnBase.getOrder(DlnOrderLib.Order)._order (contracts/DLN/DlnBase.sol#331) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#14) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#15) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._bytes (contracts/libraries/BytesLib.sol#229) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._start (contracts/libraries/BytesLib.sol#230) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._length (contracts/libraries/BytesLib.sol#231) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._start (contracts/libraries/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._start (contracts/libraries/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._start (contracts/libraries/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._start (contracts/libraries/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._start (contracts/libraries/BytesLib.sol#352) is not in mixedCase
```

```
Parameter BytesLib.toUint128(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._start (contracts/libraries/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._start (contracts/libraries/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#440) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#441) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#297-306) uses literals with too many digits:
        - tempAddress = mload(uint256)(_bytes + 0x20 + _start) / 0x1000000000000000000000000 (contracts/libraries/BytesLib.sol#302)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is never used in DlnBase (contracts/DLN/DlnBase.sol#15-341)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
getOrderId(DlnBase.Order) should be declared external:
        - DlnBase.getOrderId(DlnBase.Order) (contracts/DLN/DlnBase.sol#331-333)
getChainId() should be declared external:
        - DlnBase.getChainId() (contracts/DLN/DlnBase.sol#336-340)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- All the reentrancies flagged are false positives. The main reentrancy point in the smart contracts are the functions `_safeTransferETH()` and `_safeTransferEthOrToken()`. Every public/external function that internally calls these functions make use of the `nonReentrant` modifier.
- No major issues found by Slither.

7.2 AUTOMATED SECURITY SCAN

Description:

Halborn used automated security scanners to assist with detection of well-known security issues and to identify low-hanging fruits on the targets for this engagement. Among the tools used was MythX, a security analysis service for Ethereum smart contracts. MythX performed a scan on the smart contracts and sent the compiled results to the analyzers to locate any vulnerabilities.

MythX results:

DlnDestination.sol

Report for contracts/DLN/DlnDestination.sol
https://dashboard.mythx.io/#/console/analyses/ee9ee887-3fb9-4a90-954e-dc94a6da324d

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

DlnSource.sol

Report for contracts/DLN/DlnSource.sol
https://dashboard.mythx.io/#/console/analyses/03db8cf1-0274-4a2c-b495-fe42f2a5179b

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

DlnBase.sol

Report for contracts/DLN/DlnBase.sol
https://dashboard.mythx.io/#/console/analyses/d15cb4fa-2b9e-497d-a04c-1372559f7cc9

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

- No major issues found by MythX.

THANK YOU FOR CHOOSING

// HALBORN