tag

Noah Marconi

CTO

Who am I

12+ year research and software development background with companies such as Vision Critical, LoyaltyOne, Blockable, and Tag Innovation.

Co-designed and developed George Brown College's Blockchain Development Program. The first of its kind in Canada.

Numerous blockchain pilots and mainnet deployments.

Speaker at industry events, including Deloitte's CryptoCamp & Airline Information Events.

Four strategies

Data Fields

- Lock in what you know
- Toggle what you don't

Start Over

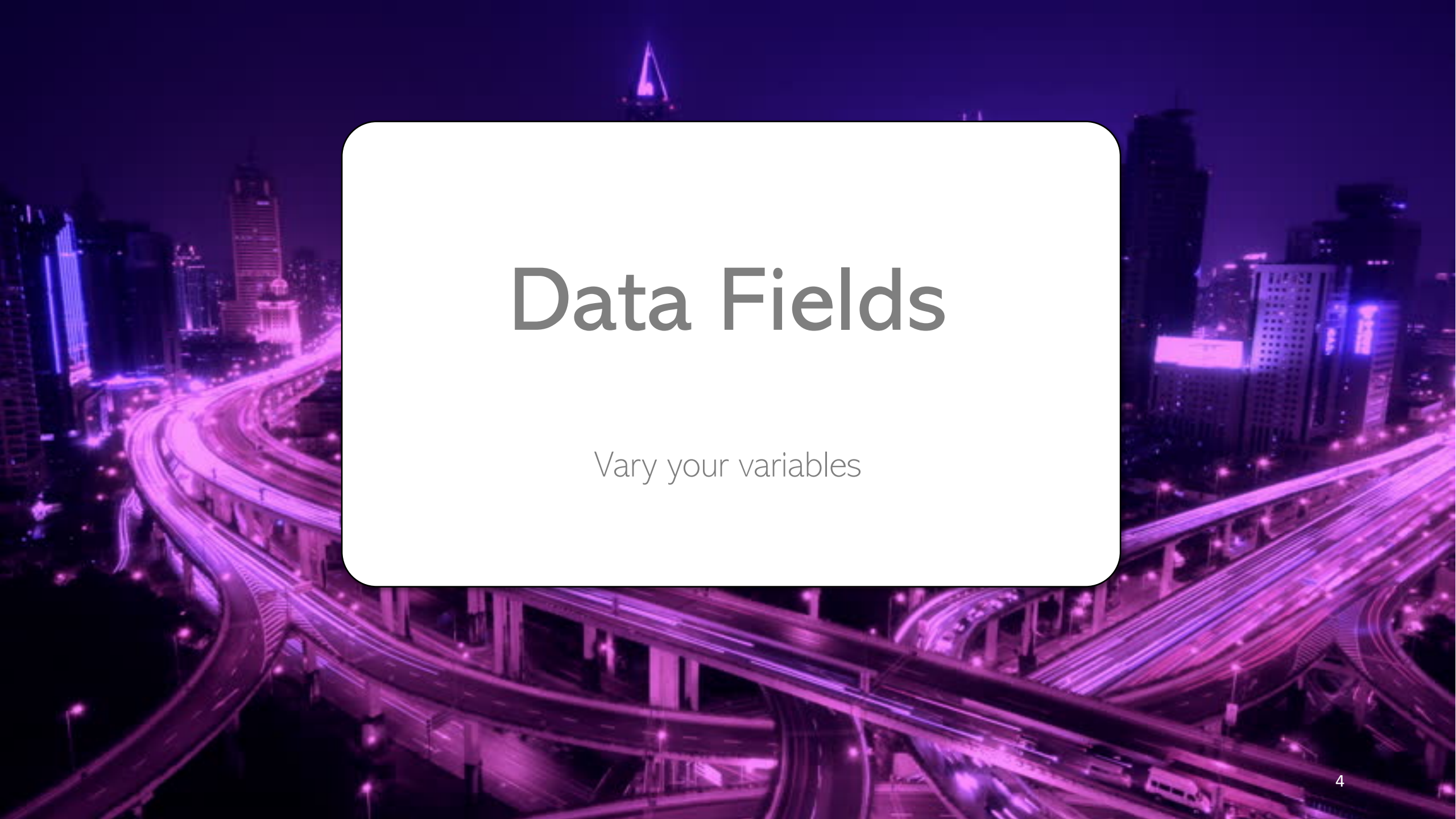
- Deploy new contract
- Migrate data over

Keep Data, New Address

- Keep data contract
- Create new logic contract
- Interact with data contract from logic contract

Keep Data & Address

- Keep data contract
- Keep contract address
- Delegate calls to new logic contract



Data Fields

Vary your variables

Scenario

- Crowdsale contract commissioned
- Requires audit ASAP
- Client's multisig address is…unknown

tag

```
/*------------ Global Address Variables ------------*/

// Recipient of the ETH funds collected.
address public multiSig;
bool public multiSigSet = false;
```

```
/*------------ Modifiers ------------*/

/**
 *
 * @dev Ensures multisig beneficiary address is set.
 *
 */
modifier multiSigIsSet() {
    require(multiSigSet);
    _;
}
```

tag

```
/*------------ Owner: Variable Setters ------------*/

/**
 * @dev Allows owner to set the multiSig (i.e. beneficiary) address.
 * To maintain trustlessness, the address can only be set one time.
 * @param _multiSig The multiSig address to receive sale proceeds.
 */
function setMultiSig(address _multiSig) public onlyOwner {
    require(!multiSigSet);
    require(_multiSig != address(0));
    require(_multiSig != address(this));

    multiSigSet = true;
    multiSig = _multiSig;
}
```

```
/*----------- Internal Methods -----------*/

/**
 * @dev Forwards funds collected to secure multiSig wallet.
 */
function forwardFunds() internal multiSigIsSet {
    multiSig.transfer(msg.value);
}
```

Scenario

- Token needs to be “issued” yesterday
- Might be vesting
- Vesting terms TBD

tag

```
interface ILocking {
    function canTransfer(address sender, uint256 value)
        external
        returns (bool);
}
```

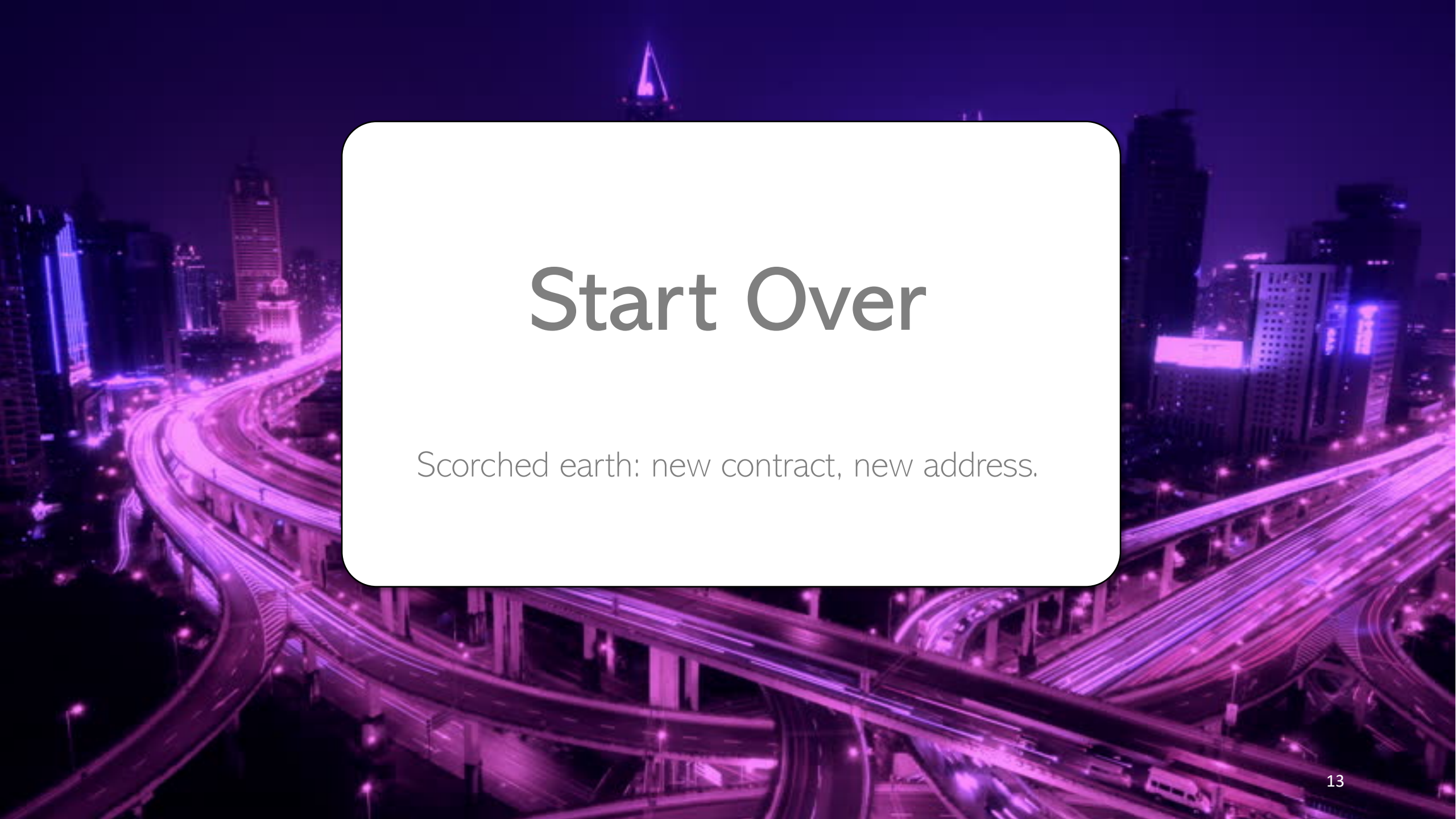
```
/*---------- Globals ----------*/

// (GMT) Thursday, July 15, 2021 0:00:00.
uint256 public constant lockEnd = 1626307200;

// Locking logic contract.
ILocking private locking;

// Disable locking logic update flag.
bool public canUpdateLockingLogic = true;
```

```
/*---------- Locking Logic ----------*/

function updateLockingContract(ILocking newLocking)
    public
    onlyMinter
{
    require(
        canUpdateLockingLogic,
        "Locking contract updates are disabled.”
    );
    locking = newLocking;
}

function disableLockingLogicUpdate()
    public
    onlyMinter
{
    canUpdateLockingLogic = false;
}
```

tag

tag

```
/*---------- Transfer overrides ----------*/

function transfer(address to, uint256 value)
    public
    returns (bool)
{
    // solium-disable-next-line security/no-block-members
    if(block.timestamp < lockEnd && locked[msg.sender]) {
        require(
            locking.canTransfer(msg.sender, value),
            "Invalid transfer. Funds still locked."
        );
    }
    return super.transfer(to, value);
}
```



Start Over

Scorched earth: new contract, new address.

Scenario

- Launched a token
- Errors :(

Scenario

- Launched a token
- Errors :(
- We were prepared :)

tag

```
interface MigrateContract {
    function migrateData(address account, uint256 amount) external;
}
```

```
/*---------- Types ----------*/

enum Stage { Active, Stopped, Migrating }

/*---------- Globals ----------*/

mapping (address => uint256) public _balances;
Stage public stage;
MigrateContract private _newContract;

/*---------- Modifiers ----------*/

modifier whenLive() {
    require(
        stage == Stage.Active,
        "Function not permitted when not Status.Active"
    );
    _;
}
```

tag

```
/*---------- Owner Methods ----------*/
function retireContract()
    public
    whenLive
    onlyOwner
{
    stage = Stage.Stopped;
}

function initMigrate(MigrateContract newContract)
    public
    onlyOwner
{
    require(
        stage == Stage.Stopped,
        "Stage must be set to Stopped before initializing migration."
    );
    stage = Stage.Migrating;
    _newContract = newContract;
}
```

tag

```
function migrateData(address account)
    public
{
    require(
        stage == Stage.Migrating,
        "Stage must be set to Migrating before migrating data."
    );
    uint256 amountToMove = _balances[account];
    delete _balances[account];
    _newContract.migrateData(account, amountToMove);
}
```

Scenario

- We don't know what we don't know
- Elegant upgrade is required

Consensys Curated: https://consensys.github.io/smart-contract-best-practices/known_attacks/

Dasp top 10: https://dasp.co/

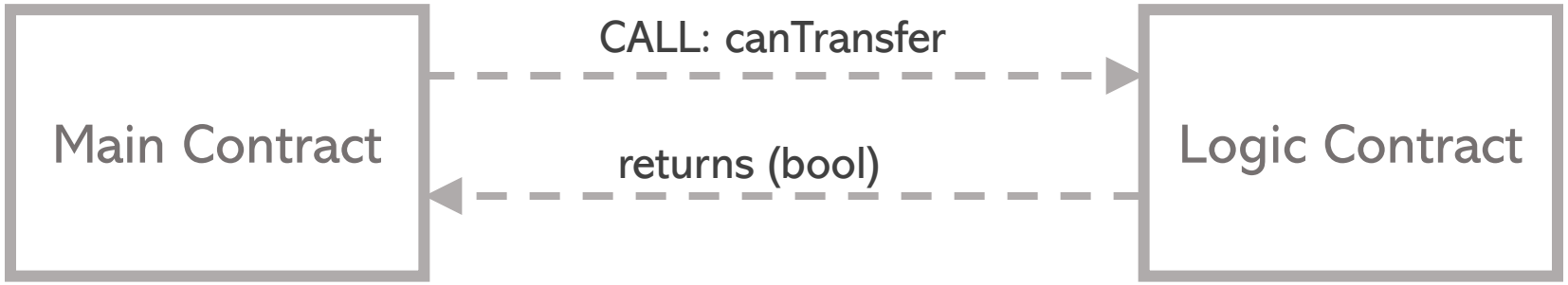
SCW Registry: https://smartcontractsecurity.github.io/SWC-registry/

`call` **vs** `delegatecall`

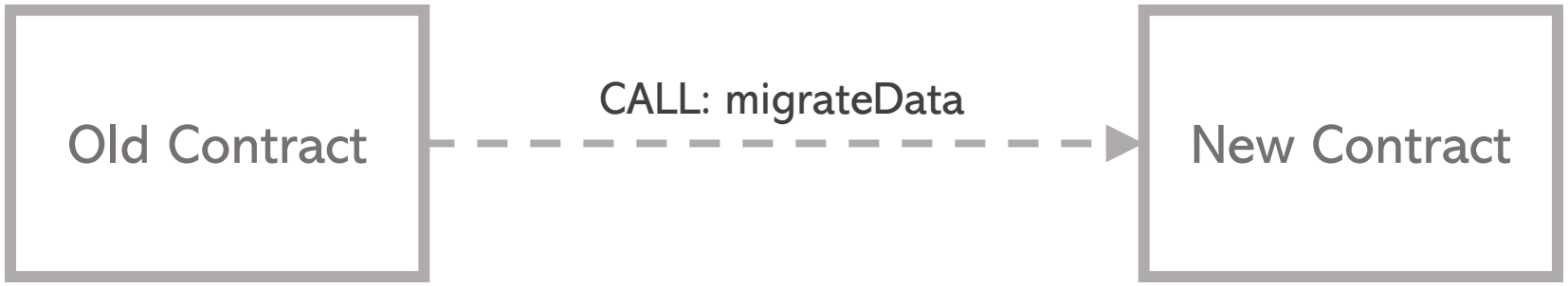
"`call` - Execute code of another contract
`delegatecall` - Execute code of another contract, but with the state(storage) of the calling contract."

Source: https://zupzup.org/smart-contract-interaction/

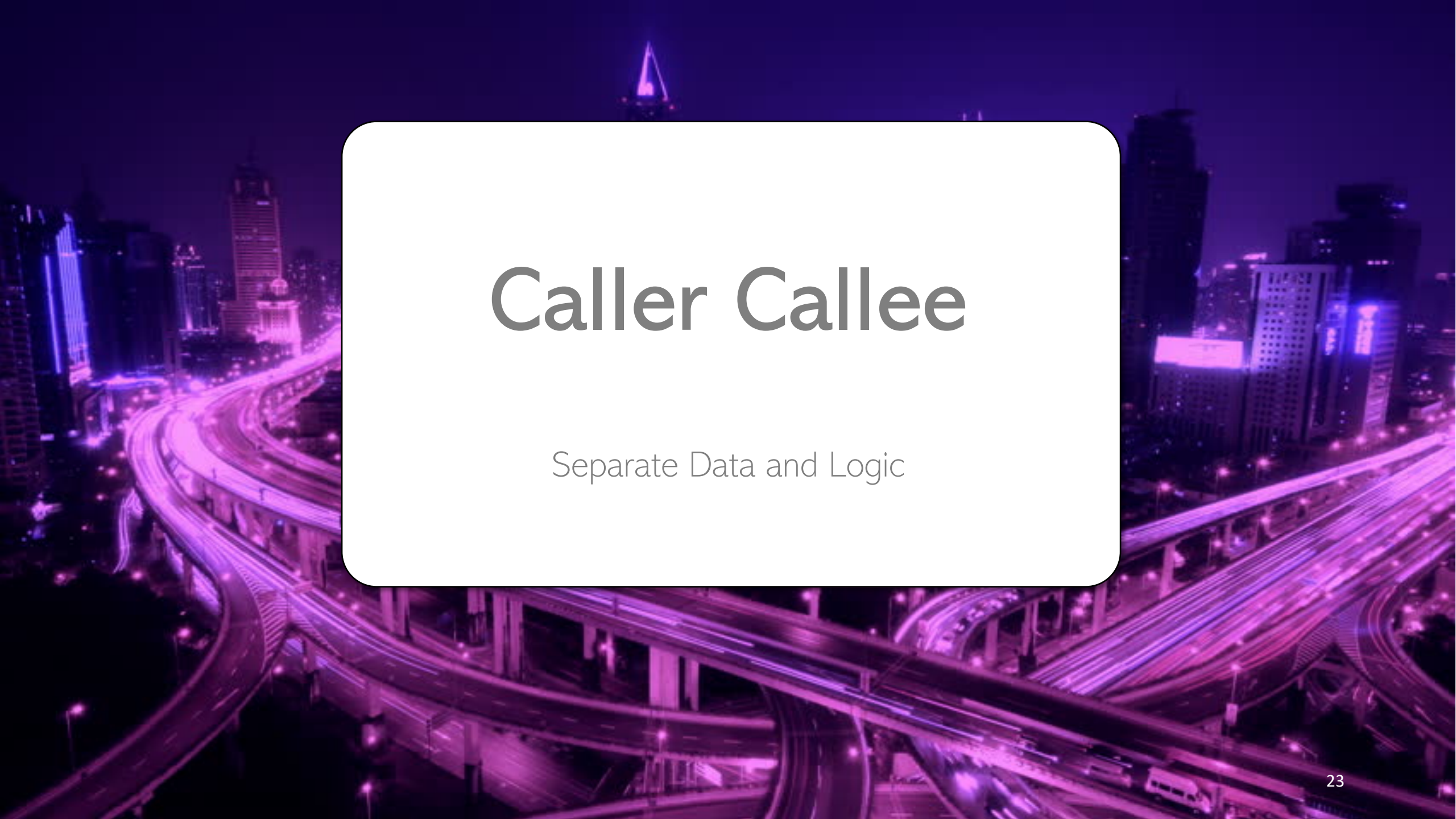
Locking Example external CALL



Migrate Example external CALL



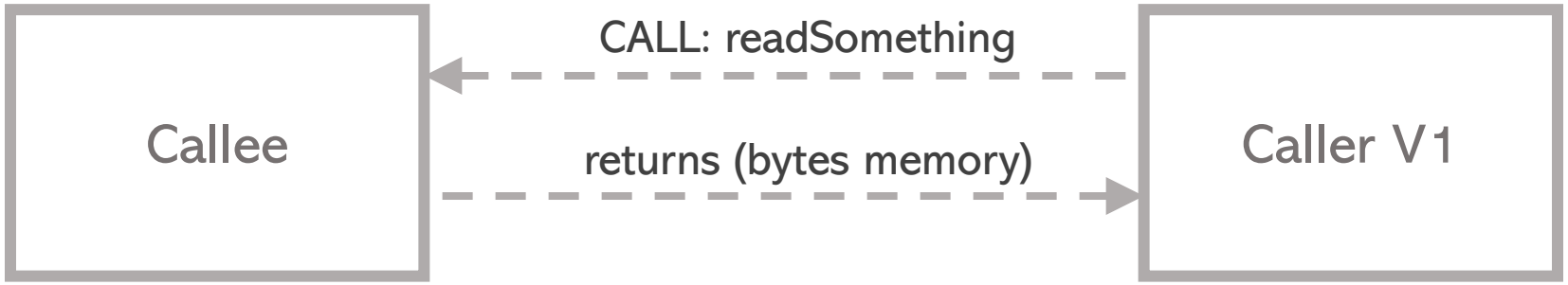
Record migrated data



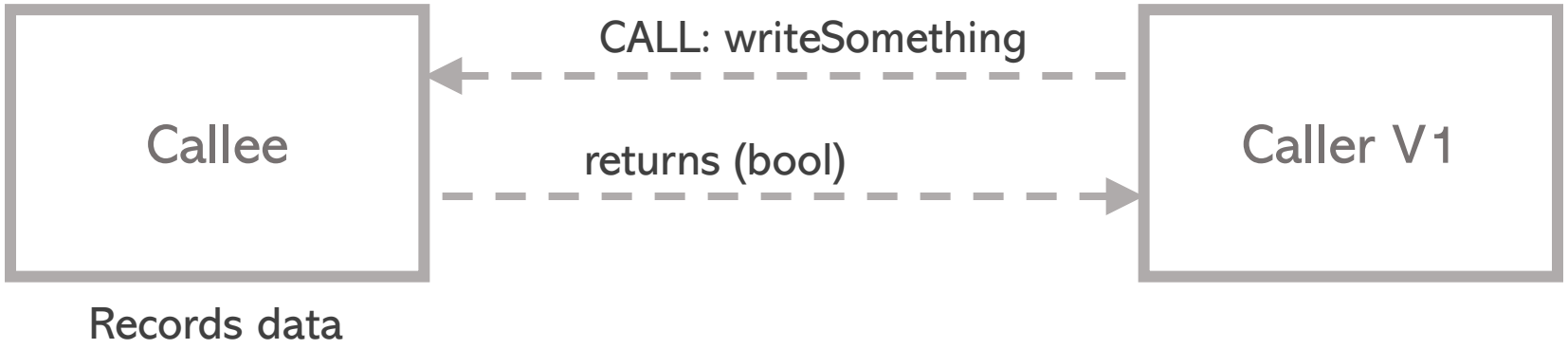
Caller Callee

Separate Data and Logic

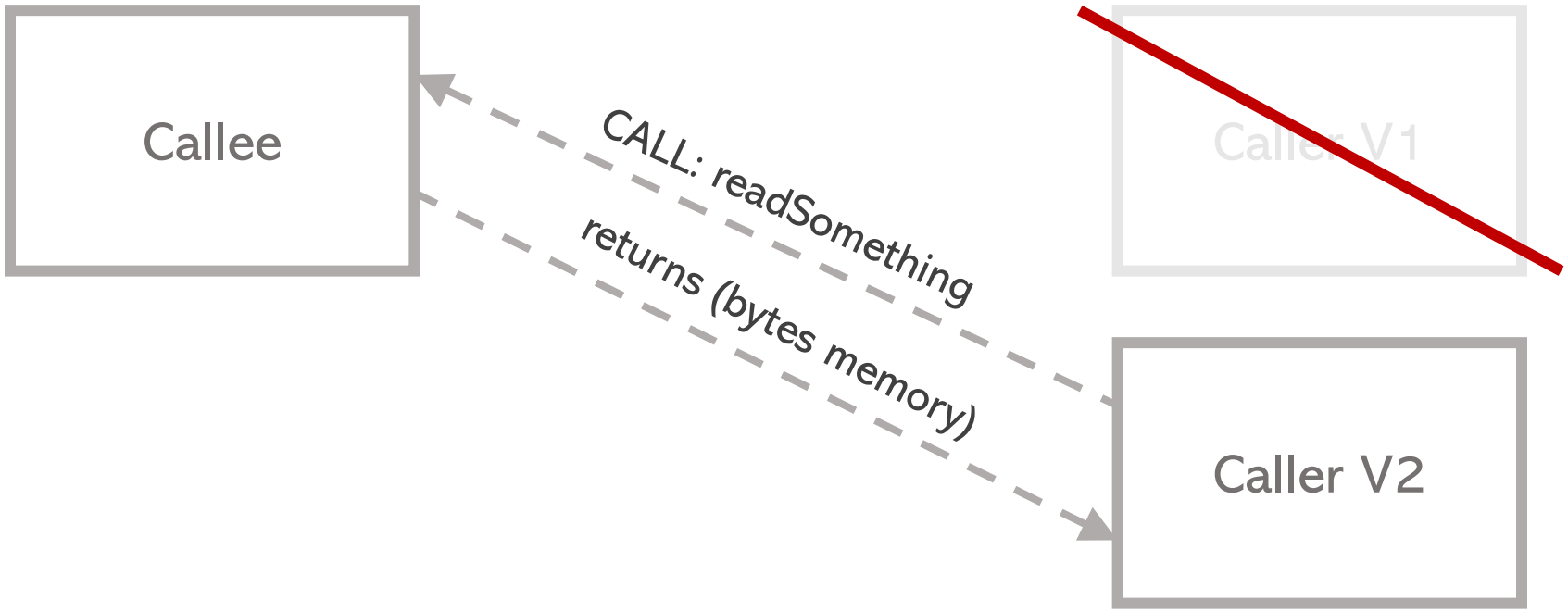
Caller Callee



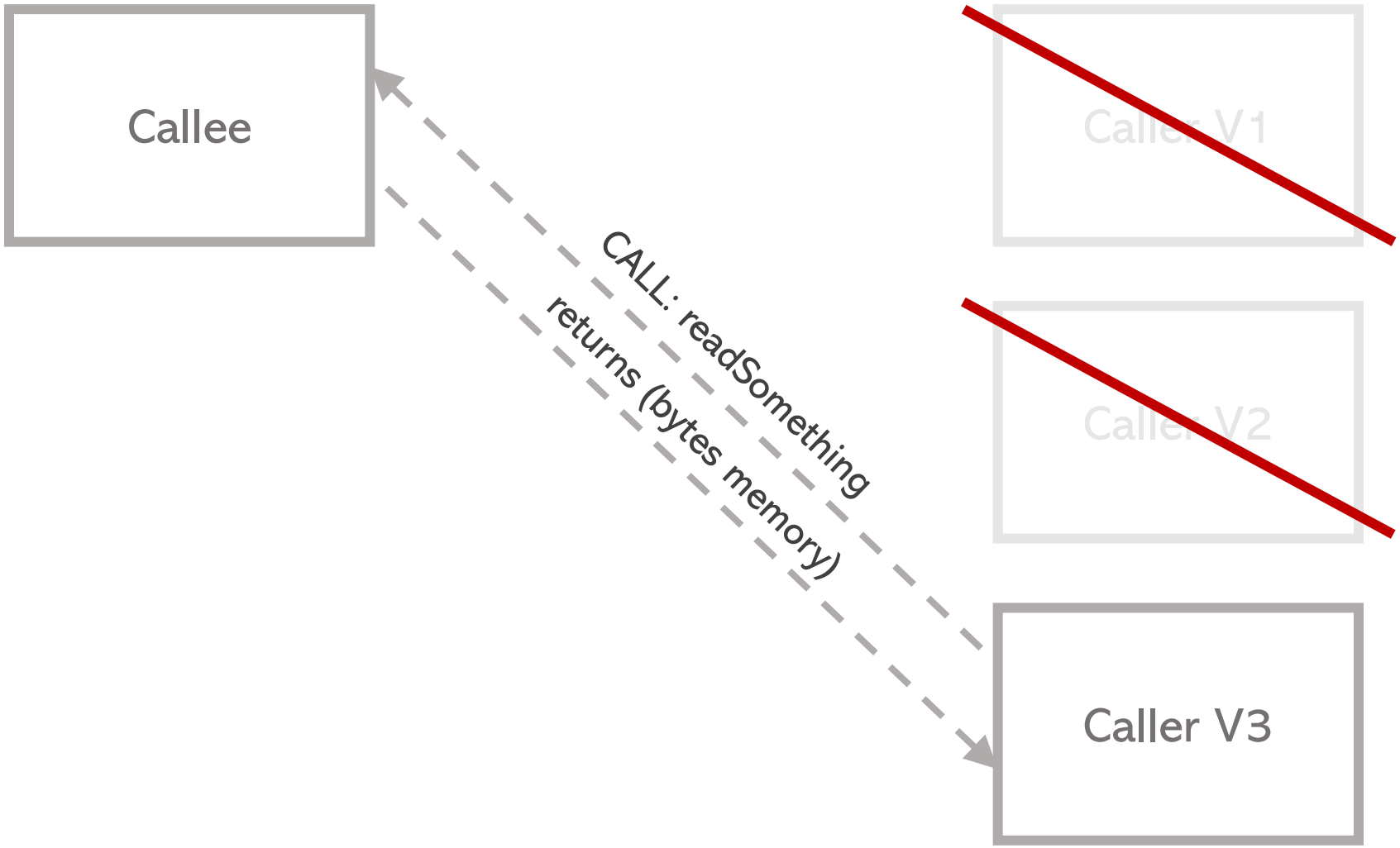
Caller Callee



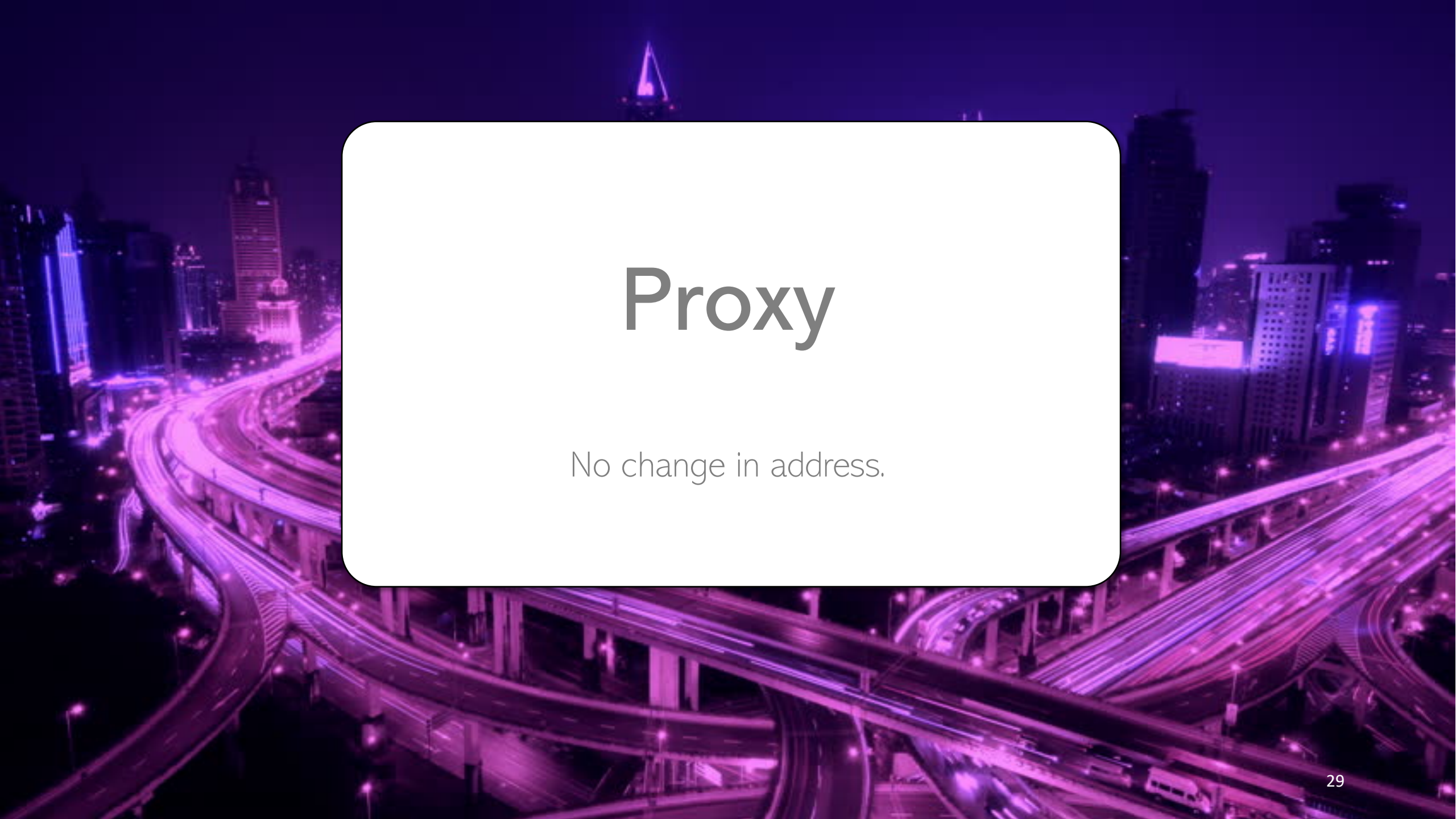
Caller Callee



Caller Callee



```
interface Callee {
    function storeBytes(bytes32 key, bytes calldata value) external;
    function getBytes(bytes32 key) external returns(bytes memory);
}

contract Caller {
    Callee public _callee;

    constructor(Callee callee) public {
        _callee = callee;
    }

    function storeBytes(bytes32 key, bytes memory value) public {
        _callee.storeBytes(key, value);
    }

    function getBytes(bytes32 key) external returns(bytes memory) {
        return _callee.getBytes(key);
    }
}
```



Proxy

No change in address.

Proxy Pattern

Source: https://blog.zeppelinos.org/proxy-patterns/

```
address _logicContract;

function implementation() private view returns(address){
    return _logicContract;
}

function setLogicContract(address logicContract) public onlyOwner {
    _logicContract = logicContract;
}

function () external payable {
    address _impl = implementation();
    require(_impl != address(0), "Logic contract cannot be address(0)");

    assembly {
        let ptr := mload(0x40)
        calldatacopy(ptr, 0, calldatasize)
        let result := delegatecall(gas, _impl, ptr, calldatasize, 0, 0)
        let size := returndatasize
        returndatacopy(ptr, 0, size)
        switch result
        case 0 { revert(ptr, size) }
        default { return(ptr, size) }
    }
}
```

tag

Storage Layout Matters

Logic contract writes to proxy contract and can accidentally overwrite its own location.

Trustlessness went out the window

Whoever can upgrade can introduce malicious code.

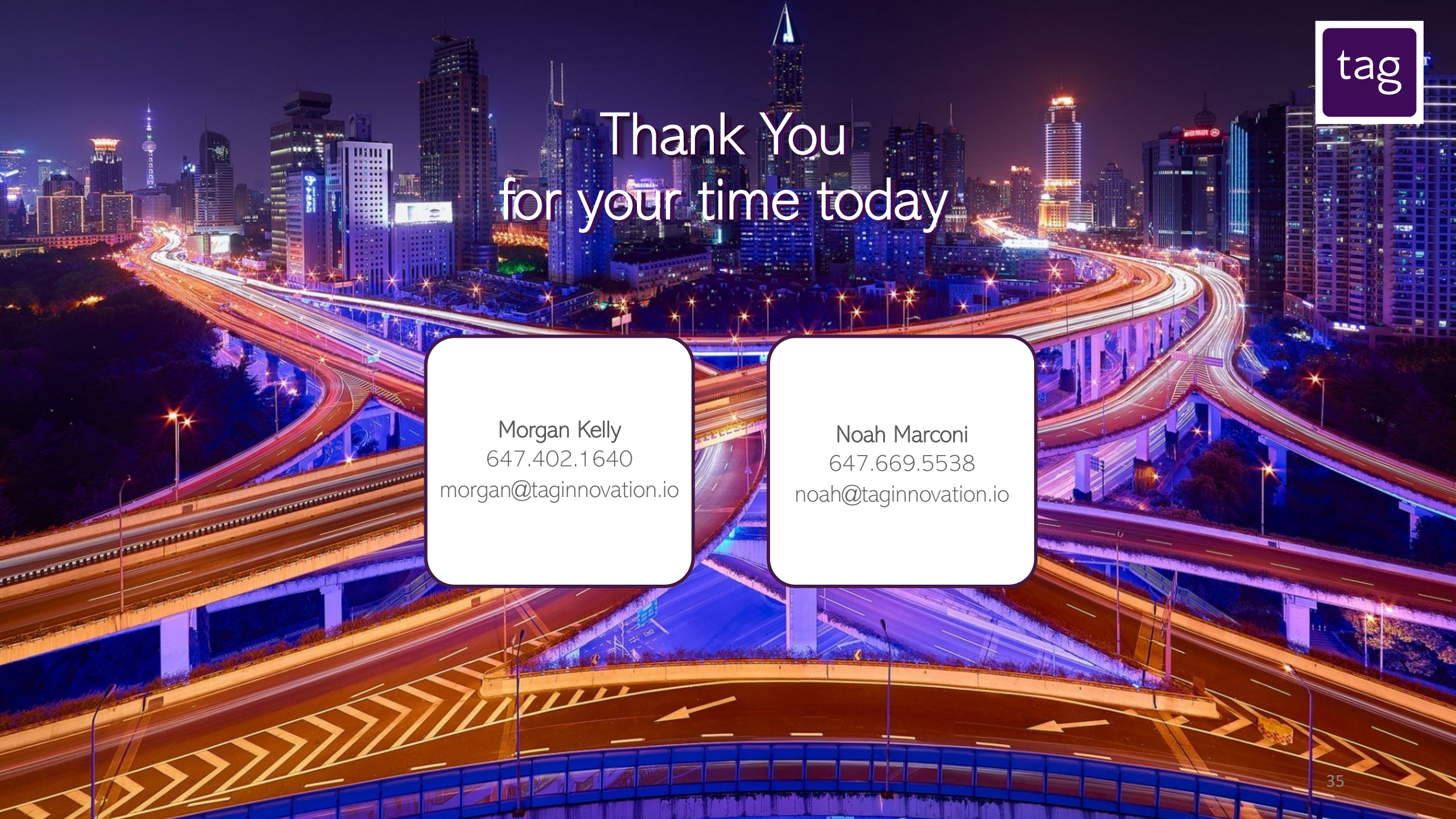
tag

Github Repo:

https://github.com/NoahMarconi/upgradeStrategies

CAUTION:

Examples meant to communicate concepts
and are not appropriate for production use.



Thank You
for your time today

Morgan Kelly
647.402.1640
morgan@taginnovation.io

Noah Marconi
647.669.5538
noah@taginnovation.io

tag