The Standards People

ETSI ISG PDL Efforts on Smart Contracts

Presented by: **Tooba Faisal**

For: **EU Commission Workshop**

07.06.2022

Agenda

- Requirements and Limitations of Smart Contracts
- Smart contracts: ETSI PDL 004, ETSI ISG PDL 011
- Designing secure smart contracts

Smart Contracts - Limitations and Requirements

Transparency

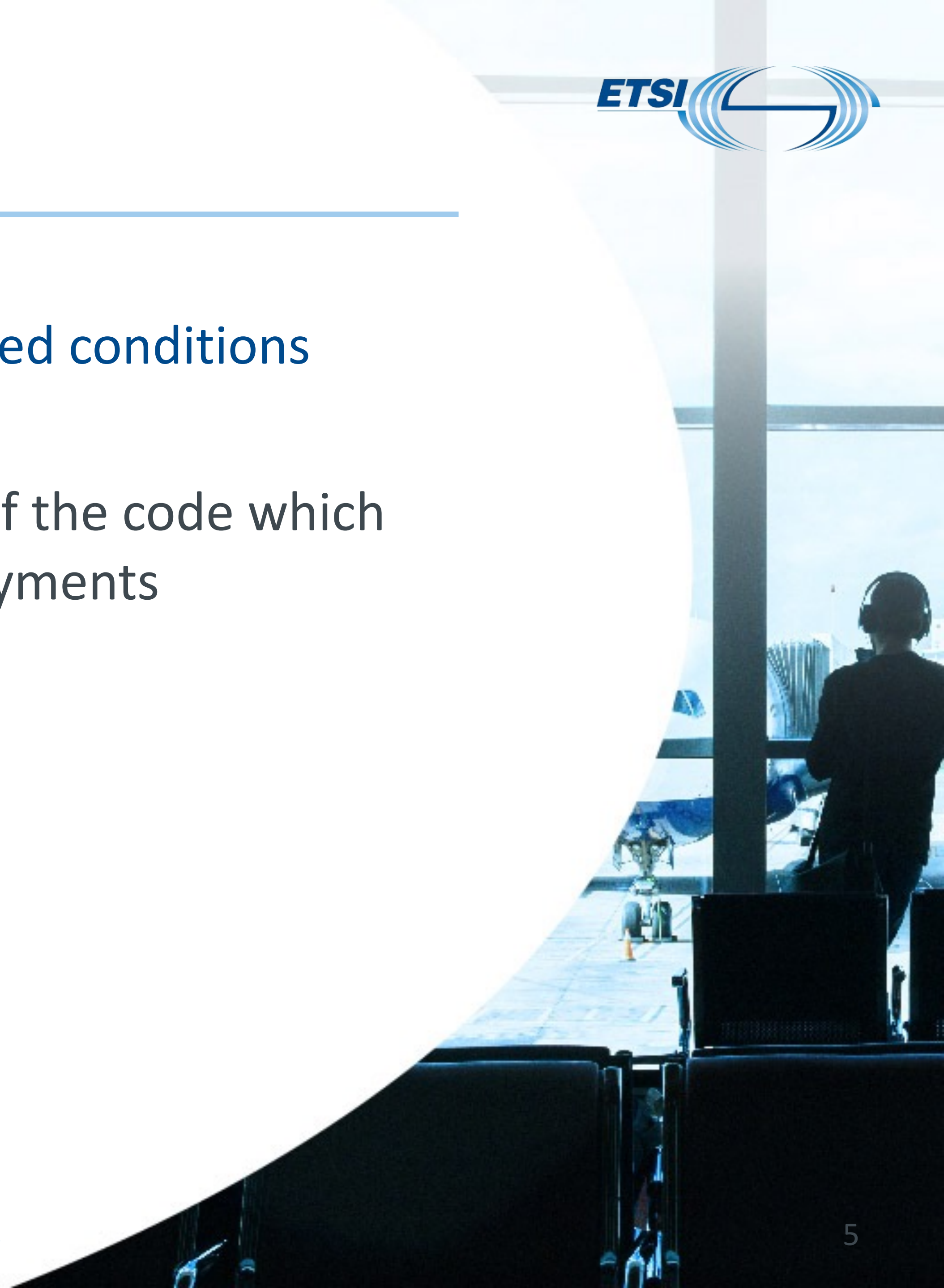
Because PDLs are transparent, smart contracts and all their respective transactions are visible to all the parties of the contract.

- Contracts are visible in a PDL, if a visibility domain is not specified, can cause contracts to be visible to unintended parties within the PDL.

Auto-executable

Smart contracts are self-executable – Pre-programmed conditions trigger these contracts.

- Erroneous code can trigger unwanted functions of the code which may cause monetary losses such as unwanted payments



Immutable

Smart contracts are immutable – because they are installed on a PDL, cannot be changed or amended:

- Smart contract can not be removed - old and dormant contracts if not secure can be dangerous
- If a smart contracts has some errors in a code it can leave back doors open- means they may be callable by unauthorized contracts

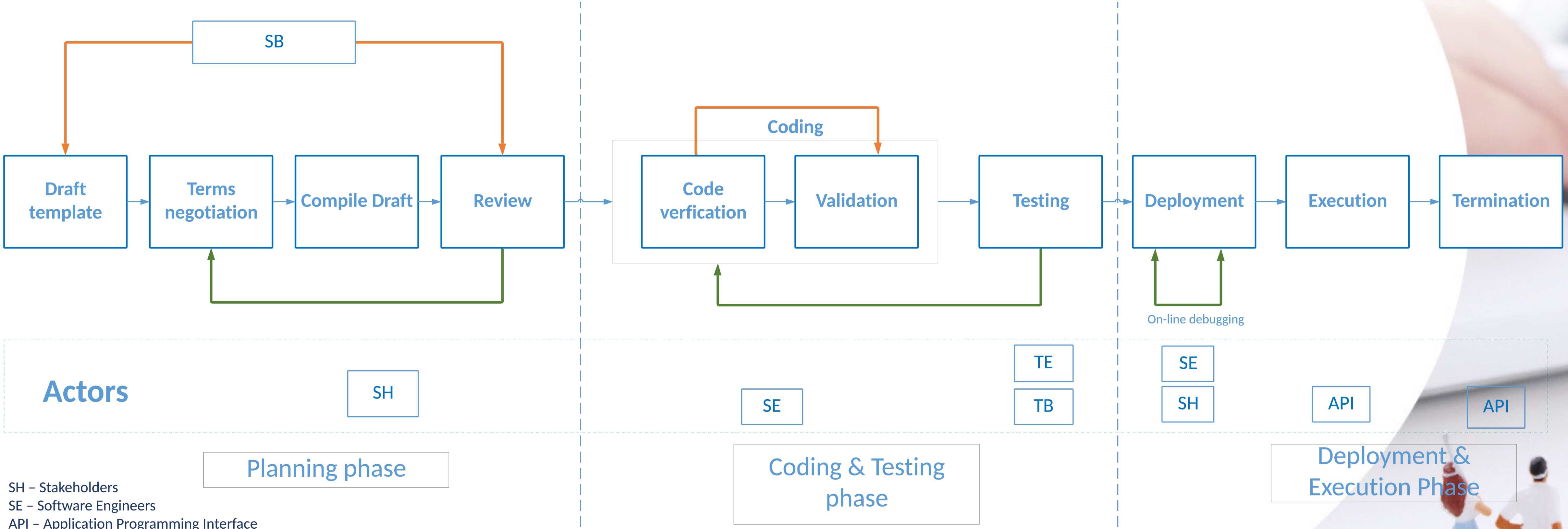
Designing secure and scalable smart contracts

Approaches taken in ETSI PDL 004 and PDL 011

To design water-tight secure contracts. In ETSI PDL 004 we discuss in detail

- Access control – a smart contract must be called by authorized parties only.
- Smart contract development life cycle – planning, development and deployment stages must be defined clearly.
- Three pass approach

Smart Contract Development Life cycle – ETSI ISG PDL 004/PDL 011

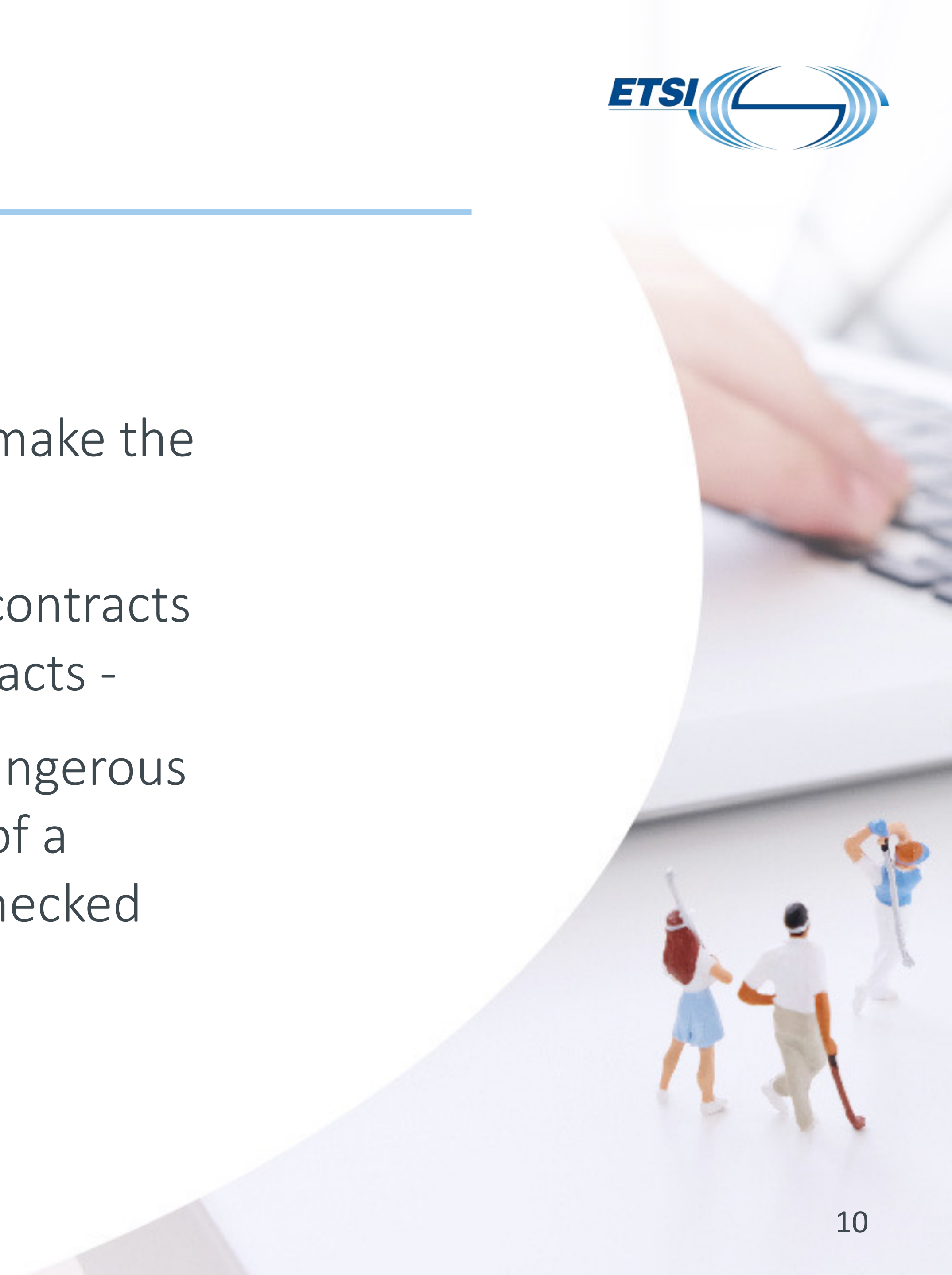


SH – Stakeholders
SE – Software Engineers
API – Application Programming Interface
TB – Test Beds
TE – Testing Engineers
SB – Standardisation Bodies

Three-Pass Approach (PDL 004)

To mitigate the dangers smart contracts posses

- Execution clauses – absence of such clauses can make the newly deployed contracts dormant
- Penetrable clauses – such clauses can cause the contracts unauthorized contracts to access the smart contracts -
- Termination clauses – Eternal contracts can be dangerous hence must be terminated exclusively. Presence of a termination clause inside the contract must be checked before deployment.

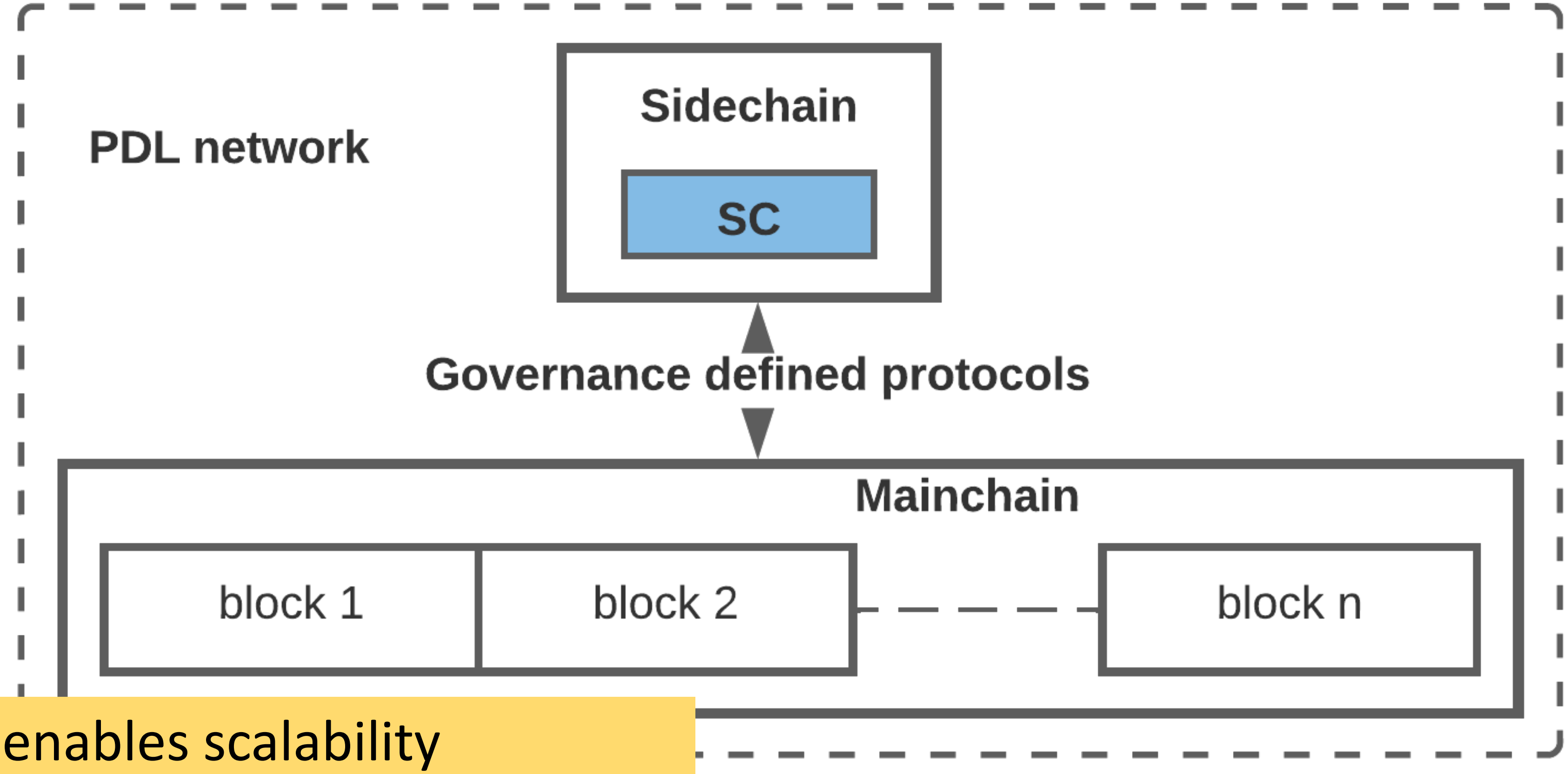


ETSI ISG PDL 011 Group Specifications Approaches

- Offloading – between sidechains, mainchains, internal and external storage
- Modularisation -- and offloading between sidechains, mainchains, internal and external storage
- Oracles – Data input should be secured
- Smart Contract Lifecycle
  - Time Limited – *Internal Timers* to avoid eternal smart contracts
  - Destruction – Smart contracts are destructed after this time
  - Access Control
  - Control Instructions
  - Strong Governance – should not confused with Trusted Third Party

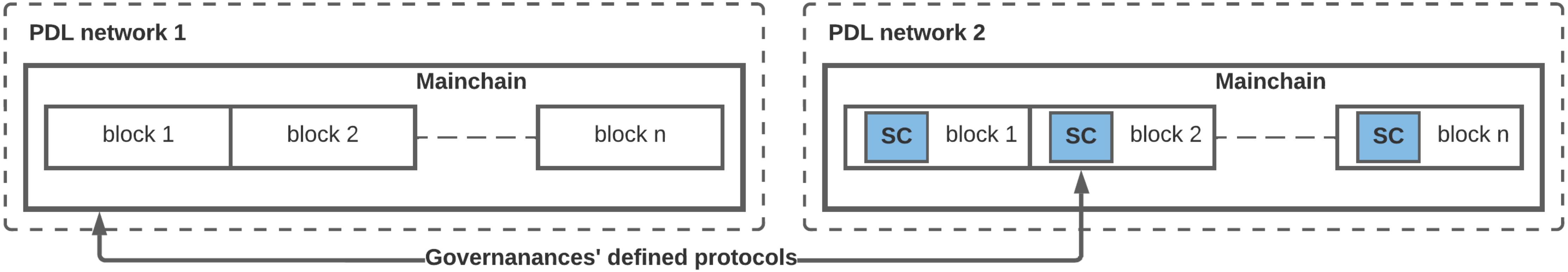
Offloading

Example - Offloading



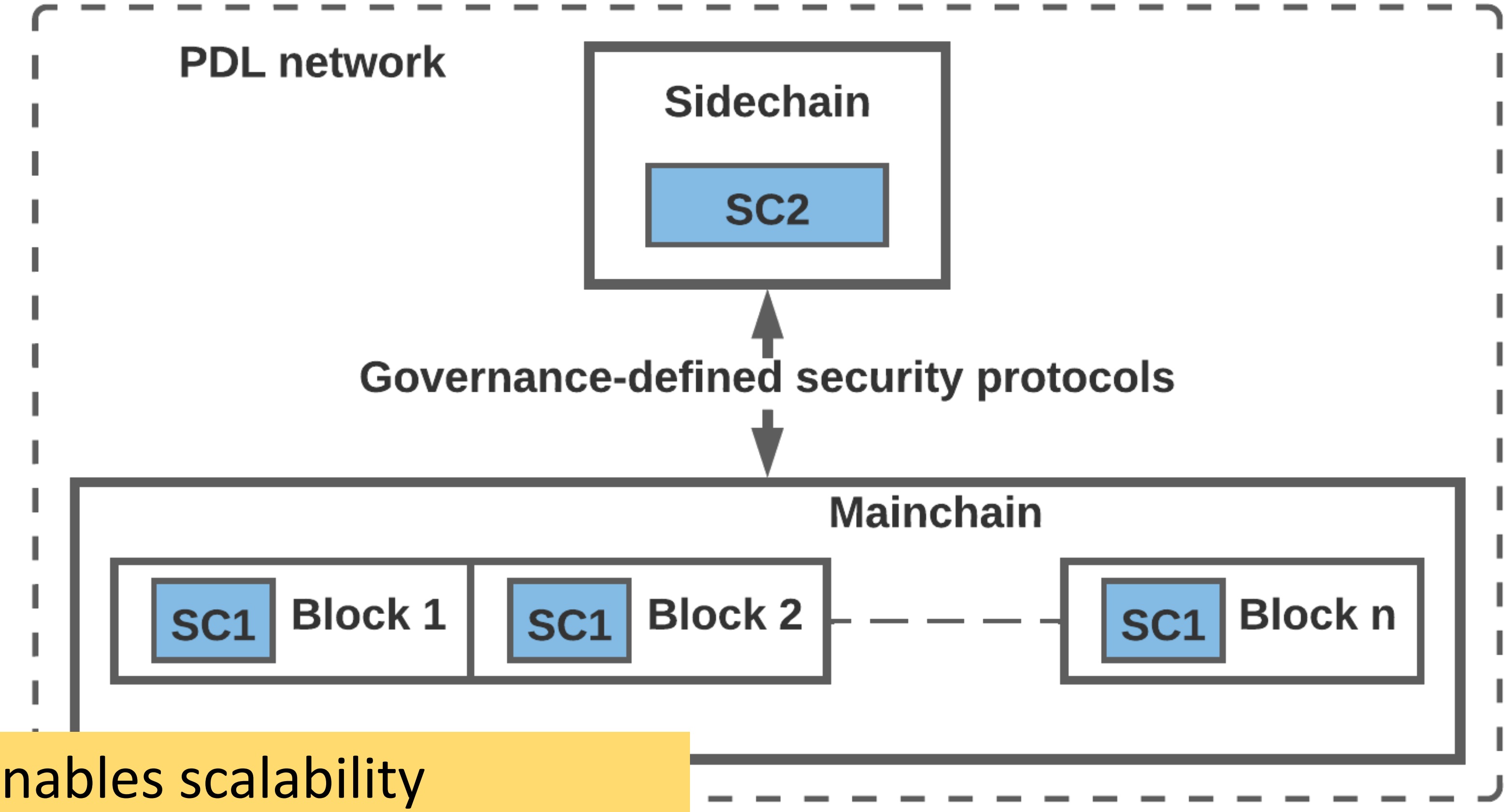
- Offloading enables scalability
- May introduce some security issues

Example -- offloading



Modularisation

Smart Contract Modularisation -- Example

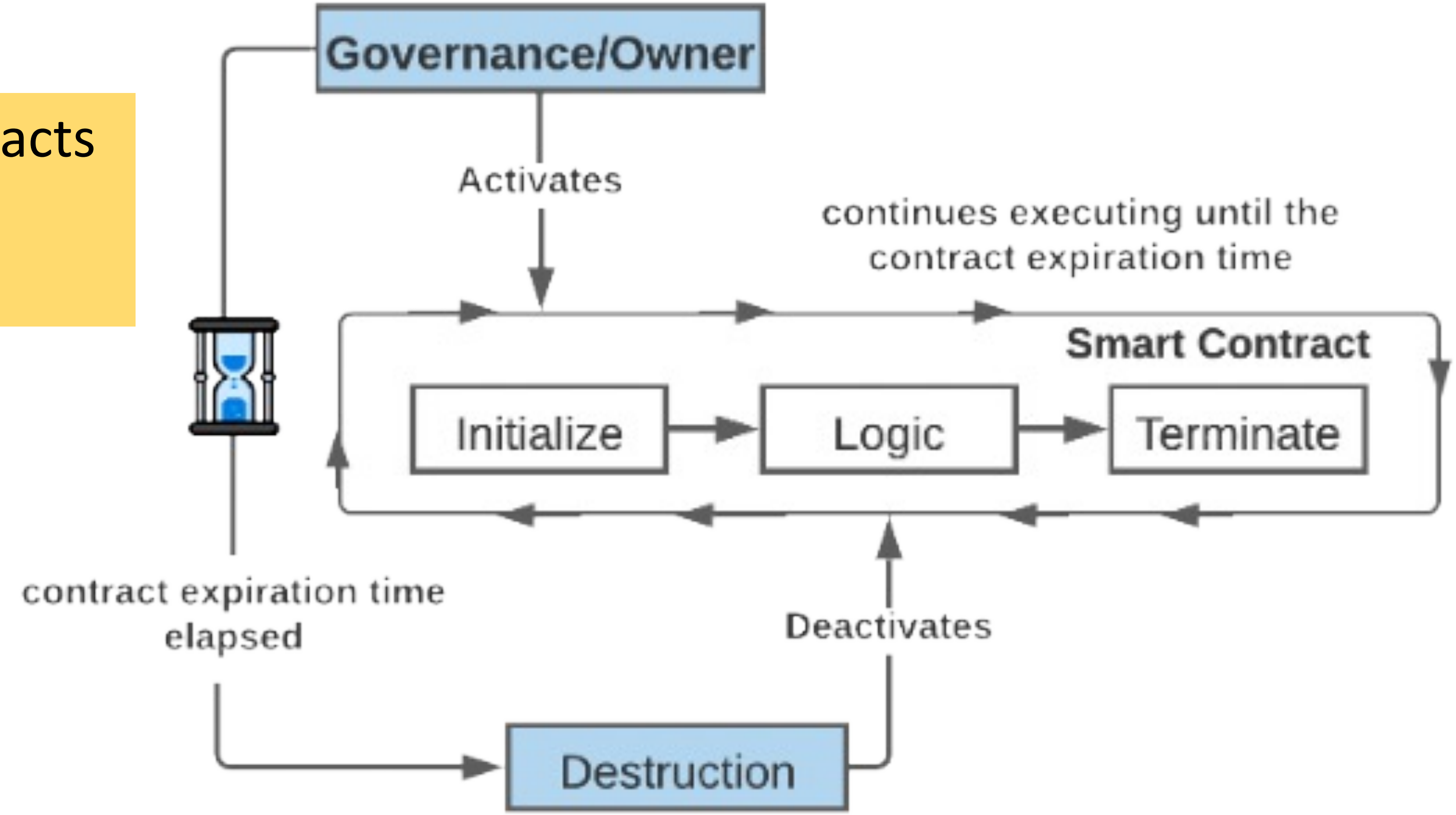


- Offloading enables scalability
- May introduce some security issues

Smart Contract Lifecycle

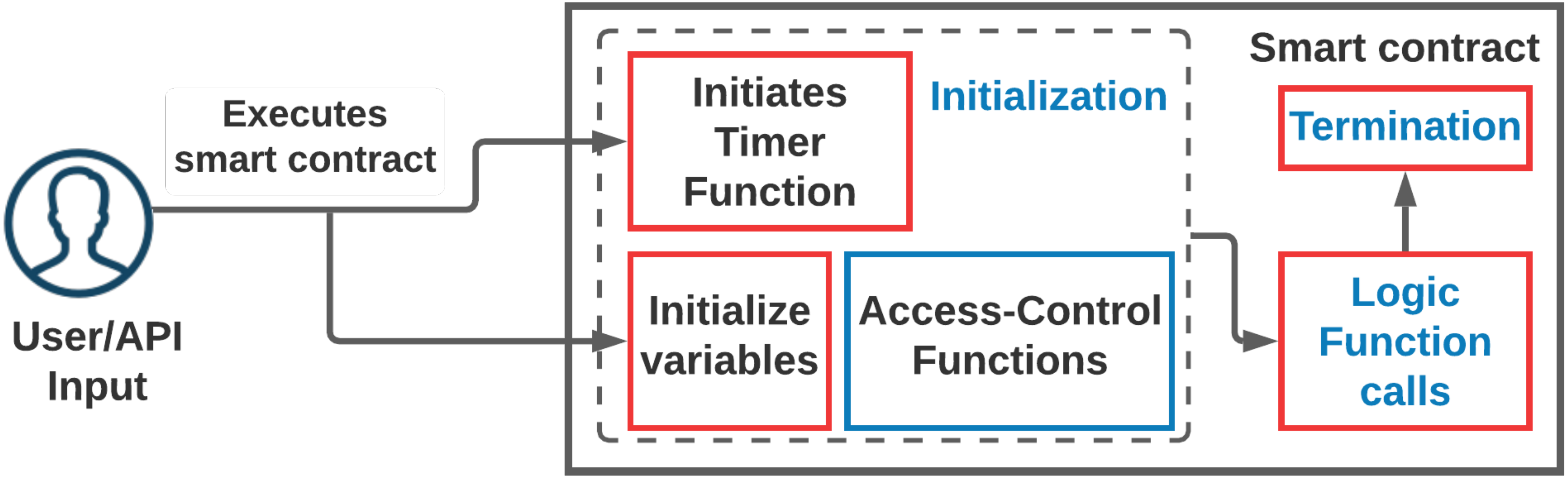
Smart Contract Lifecycle – PDL 011

Eternal smart contracts are dangerous and should not happen


Governance/Owner
Activates
continues executing until the contract expiration time
Smart Contract
Initialize
Logic
Terminate
contract expiration time elapsed
Deactivates
Destruction

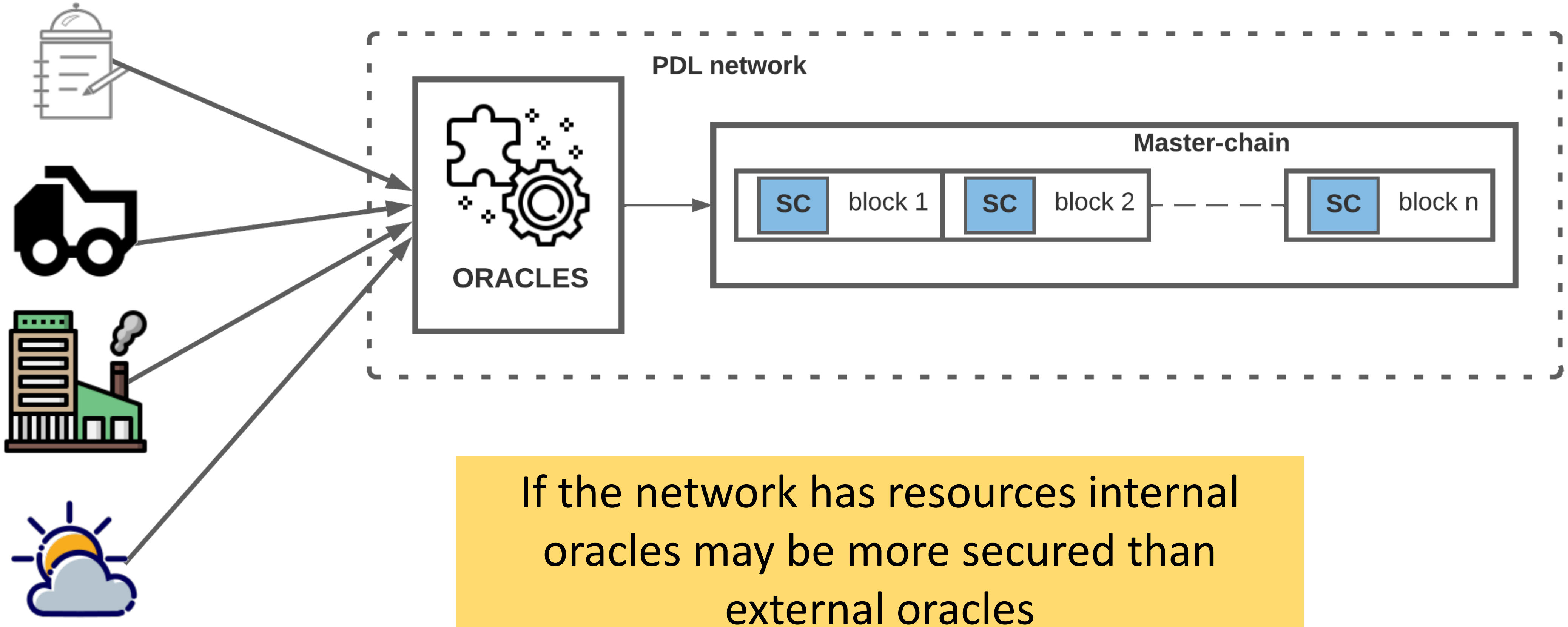
Smart Contract Architecture

Smart Contract Architecture – PDL 011



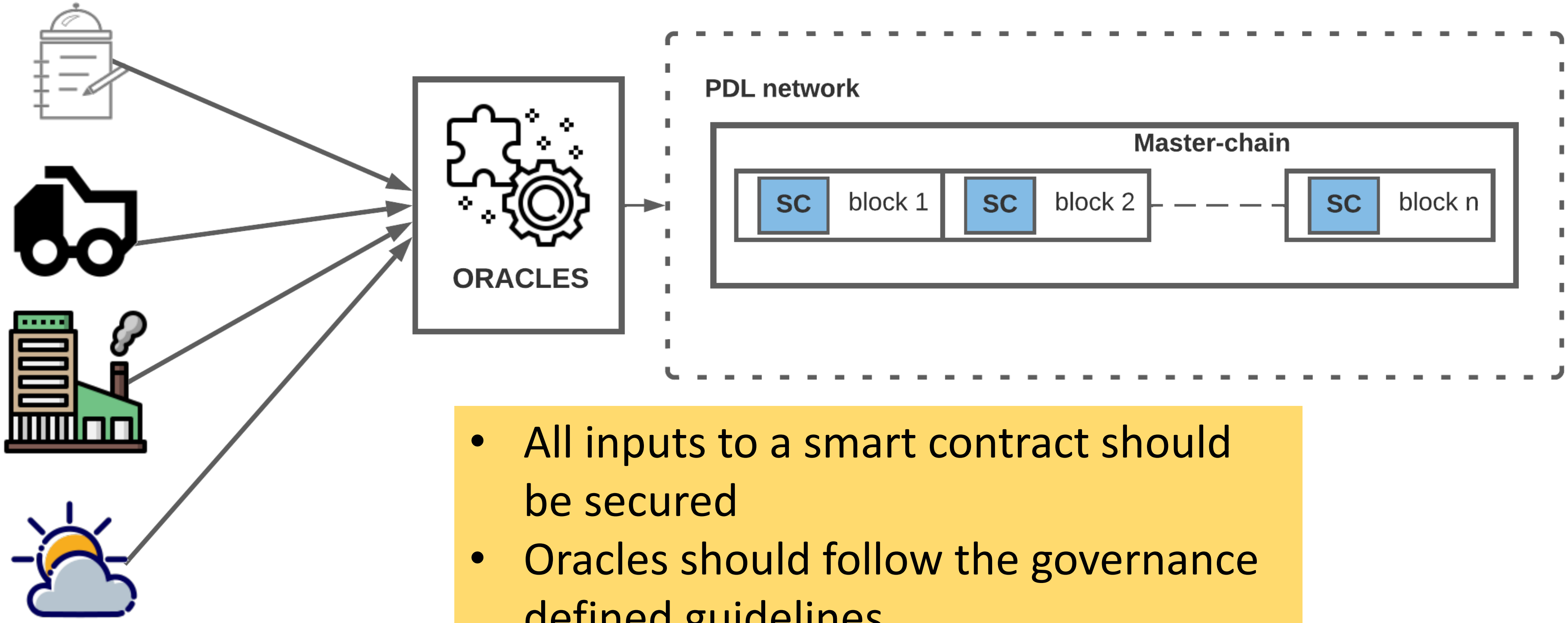
Oracles

Internal Oracles – Oracles Managed by the Governance



If the network has resources internal oracles may be more secured than external oracles

External Oracles – Oracles Managed by a Third Party



- All inputs to a smart contract should be secured
- Oracles should follow the governance defined guidelines

Conclusion

Conclusion

- Smart Contracts’ inherent properties needs to be managed
- Security of smart contracts can be achieved through careful planning
- ETSI ISG PDL is working towards designing secure and scalable smart contracts