ETSI Security Conference 2022

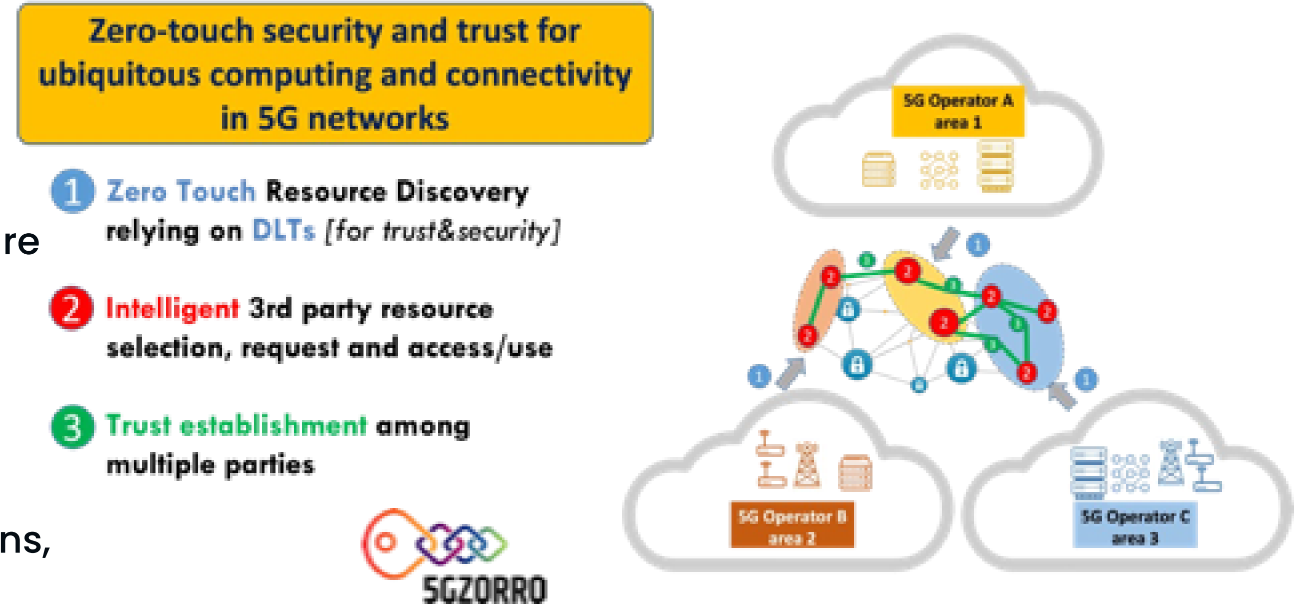
The ISG PDL Approach to Auditability and Automated Enforcement

Diego López, Telefónica (ISG PDL Chair)

05/10/2022

Some Rationale

- ICT evolution calls for automated management of Service Level Agreements (SLAs)
  - XaaS, public-private integration, multi-provider environments…
  - Enforcement of supply chain policies
  - The intent aspiration
- All parties interested
  - Providers, in improving service and infrastructure usage (and billing!)
  - Customers, in optimizing service usage and planning (and billing!)
- SLAs must be transparent to all stakeholders
  - Trustworthiness of networked services
  - Base for enhanced service consumption patterns, based on reputation and trust assessment
- A method for standardized SLA transparent recording and auditing
  - The ISG PDL proposal based on smart contracts
  - And some steps beyond



Zero-touch security and trust for ubiquitous computing and connectivity in 5G networks

1 Zero Touch Resource Discovery relying on DLTs *[for trust&security]*

2 Intelligent 3rd party resource selection, request and access/use

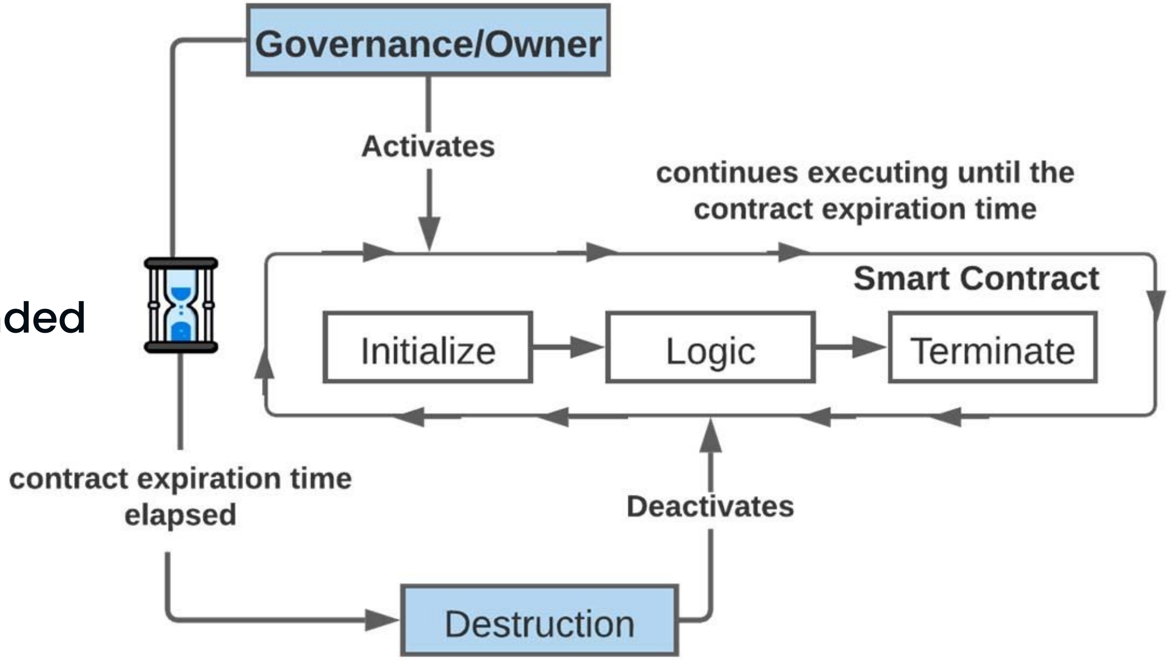
3 Trust establishment among multiple parties

The ISG PDL Goals and Scope

- Provide the foundations for the operation of (permissioned) distributed ledgers
  - Create an open ecosystem of industrial solutions
  - Deployable by different sectors
- Foster the application of the technology
  - Start from already available experiences
  - Coordinate with existing initiatives
- Define a set of well-known open operational mechanisms
  - Support their demonstration
  - Facilitate interoperability assessment
- Focus on permissioned ledgers
  - Several advantages (legal, fairness, resource consumption…) for actual industrial applications
  - But not exclusively

Introducing Smart Contracts

- Software code fragments installed on Distributed Ledgers (DLs)
  - Different storage modes
  - A defined lifecycle
- Immutable
  - Once recorded cannot be changed or amended
- Auto-executable
  - Triggered by software condition(s)
- Transparent
  - All the participants of the ledger keep the same copy

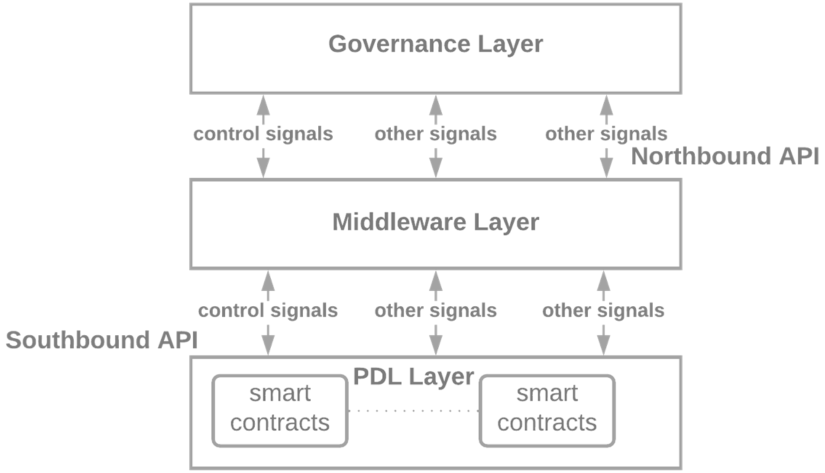

Governance/Owner
Activates
continues executing until the contract expiration time
Smart Contract
Initialize
Logic
Terminate
contract expiration time elapsed
Deactivates
Destruction

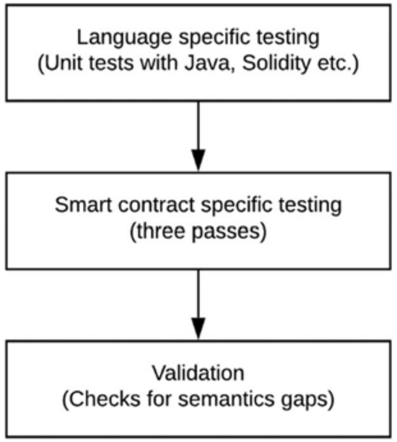
- PDL 004 provides a base architecture
- PDL 011 defines requirements on architecture and security

Inherent Properties (and Implications)

- Smart contracts are immutable – Cannot be changed or amended (as the DL itself)
  - Old and dormant contracts may extend the threat surface
  - Unattended back doors open: Calls by unauthorized contracts/actors
- Smart contracts are self-executable – Pre-programmed conditions trigger them
  - Erroneous code can trigger unwanted function
- Smart contracts are transparent - Visible to all parties
  - A visibility domain has to be specified
  - More natural in permissioned approaches

- Define specific governance layering and verification mechanisms



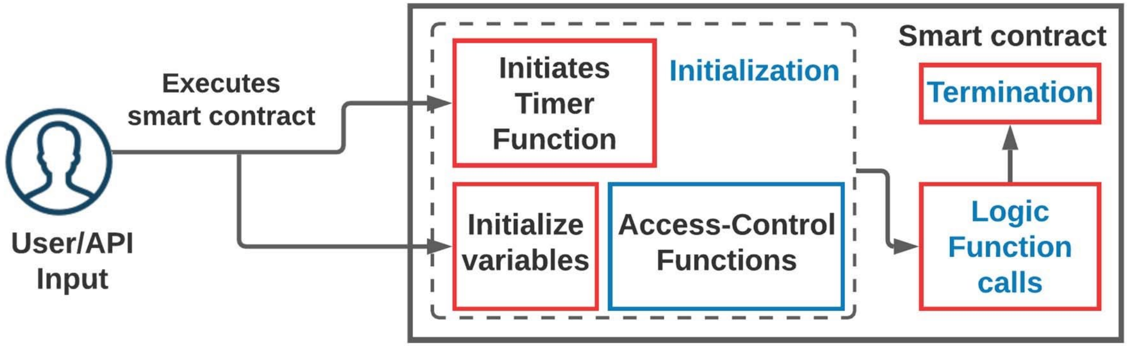


Language specific testing
(Unit tests with Java, Solidity etc.)
Smart contract specific testing
(three passes)
Validation
(Checks for semantics gaps)

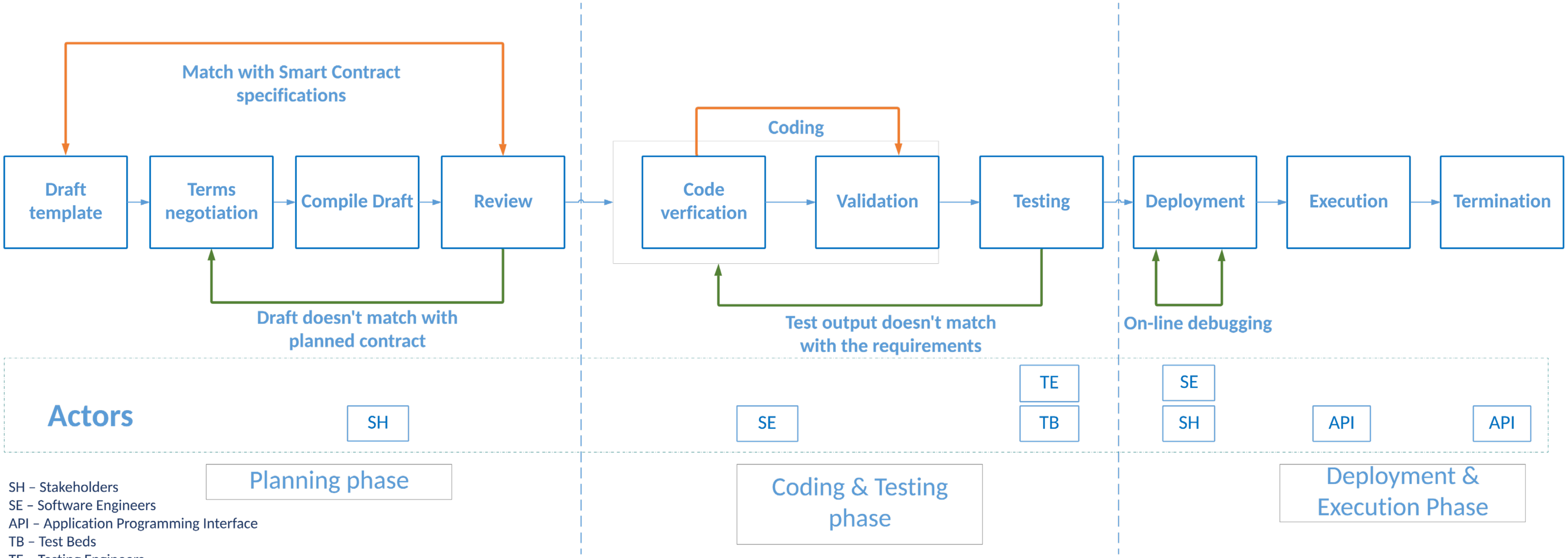
Specific Requirements

As described and analyzed in PDL 011

- Watertight Security
  - Only authorized users can access smart contracts (and specific functions only)
- Terminatable
  - Ensure that smart contracts are terminated after certain time to avoid eternal contracts.
- Auditable
  - The stakeholders should always be able to audit smart contracts' code and their libraries
    - An argument for open-source
- Upgradable
  - Follow precise versioning and safely deactivate older contracts
- Reusability
  - Cope with scalability



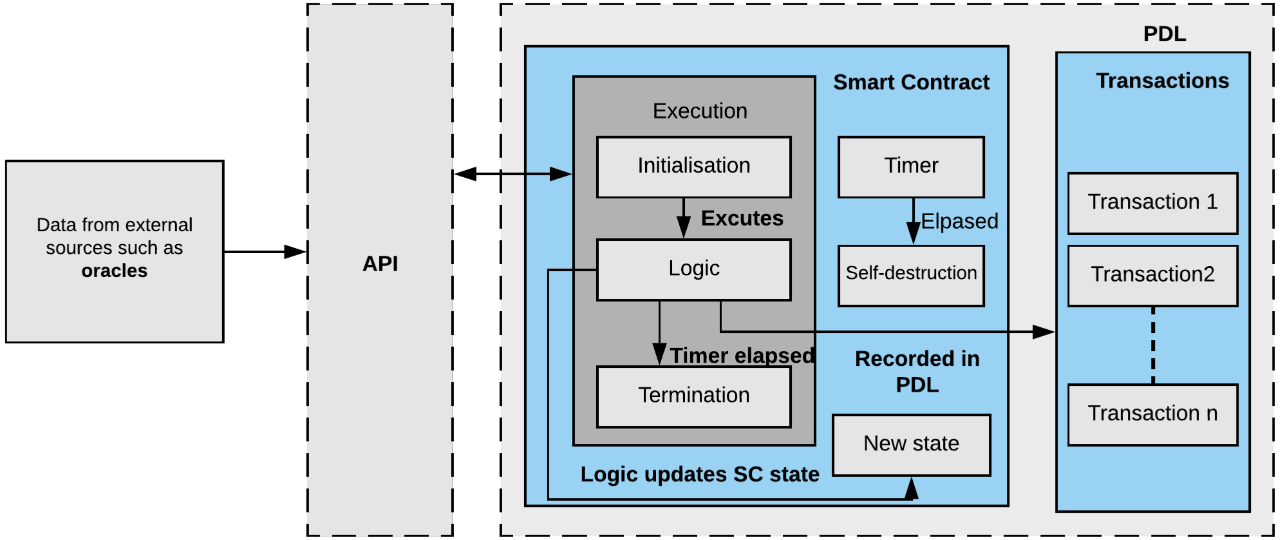
An E2E Development Lifecycle (PDL 004)



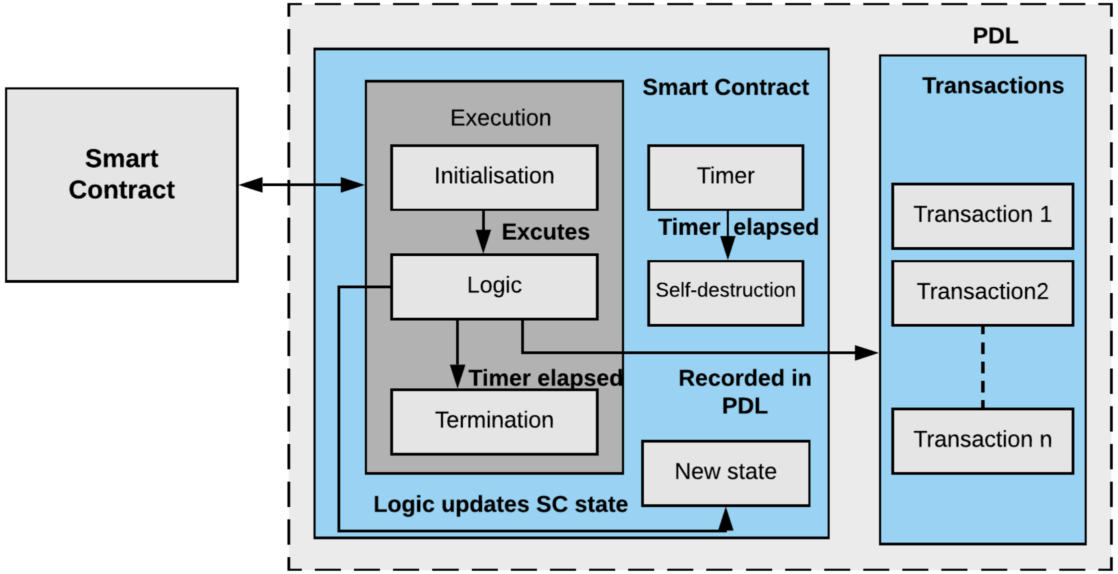
SH – Stakeholders
SE – Software Engineers
API – Application Programming Interface
TB – Test Beds
TE – Testing Engineers
SB – Standardisation Bodies

- Three main phases: planning, coding and testing, and deployment and execution
- Actors in each phase
- Explicit termination

Two Execution Models (PDL 004 & PDL 011)



- Data input exclusively by external oracles



- Smart contract composition

Related and Ongoing Work

In ISG PDL (and beyond)

- PDL 006 – Inter-ledger interoperability
  - Code access models and contract interaction
- PDL 010 – Operations in offline mode and
  PDL 020 – Wireless consensus
  - Common situations in many industrial application scenarios
- PDL 014 – Non-repudiation techniques
  - Enabling full and transparent auditability
- PDL 016 – Smart contracts and oracles
  - Issues pertaining data flows
- PDL 018 – Redactable ledgers
  - Dealing with mutability
- PDL 019 – Trust management
  - At the base of contract lifecycle

- Proposal for an EN on smart contracts
  - Joint work between ETSI (TC ESI) and CEN/CENELEC
  - Using PDL 011 as foundation

To Conclude

ISG PDL is actively working in standardizing smart contracts to address

- Inherent immutability, auto-execution and transparency
  - Well-known and open methods for verification
  - Well-established governance practices
  - Well-defined data conduits and triggers
  - Considering redactable approaches
- Their application for transparent and autonomous contracting in next-generation ICT services
- Their use to provide traceable audit mechanisms
- The assessment of the impact of ledger characteristics: consensus protocol, APIs…
- The necessary alignment of ancillary elements