

Smart Contracts

Architecture and Requirements

Presentation Plan

- Implementation Scope in Airlines Industry
- High Level Architecture
- Development Environments
- Applying Business Logic with Smart Contracts
- Proof of Concepts

Implementation Scope in Airlines Industry

The following are some of the processes which we can move to a Smart Contract based system:

Case Processing

Smart Contracts allow the performance of credible transactions without third parties.

Maintenance/VM

Transform maintenance logs for better management.

Ensure that parts procured are legitimate and can offer a “virtual copy”, including all details.

Ticketing Process

Using smart contracts associated with tickets, airlines can add business logic and terms & conditions around how the tickets to be sold and used.

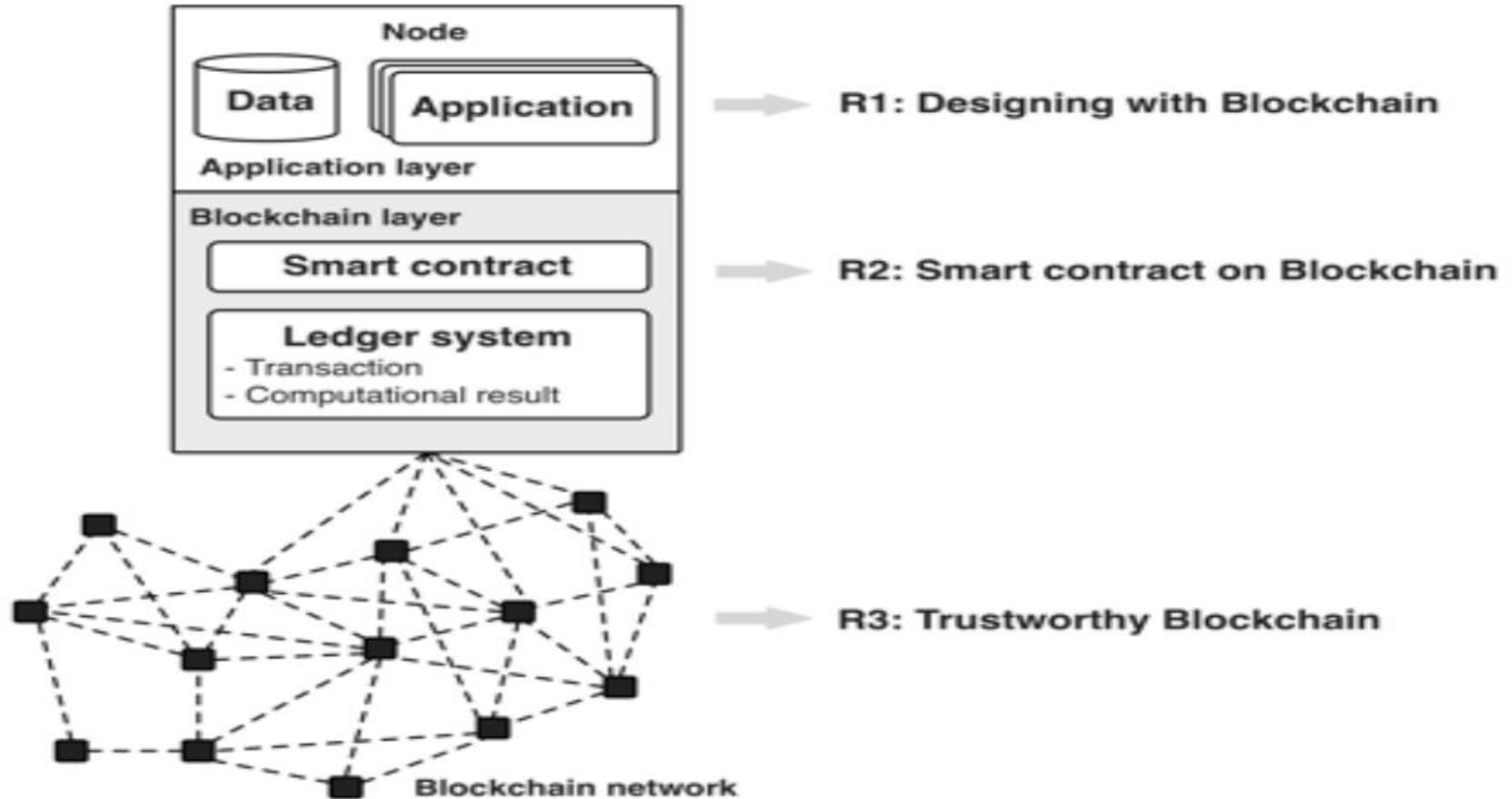
Loyalty Points

Travelers can get instant value by redeeming them on the spot instead for waiting for a long time to get points settled and accrued to use them.

*We talk about the architecture and requirements in more details in the remaining slides for implementing Case Processing process to a Smart Contracts system as it is closely related to **PNR Case Processing** process. The same methodology can be implemented across other processes with some changes/adjustments.*

High Level Architecture

Overview

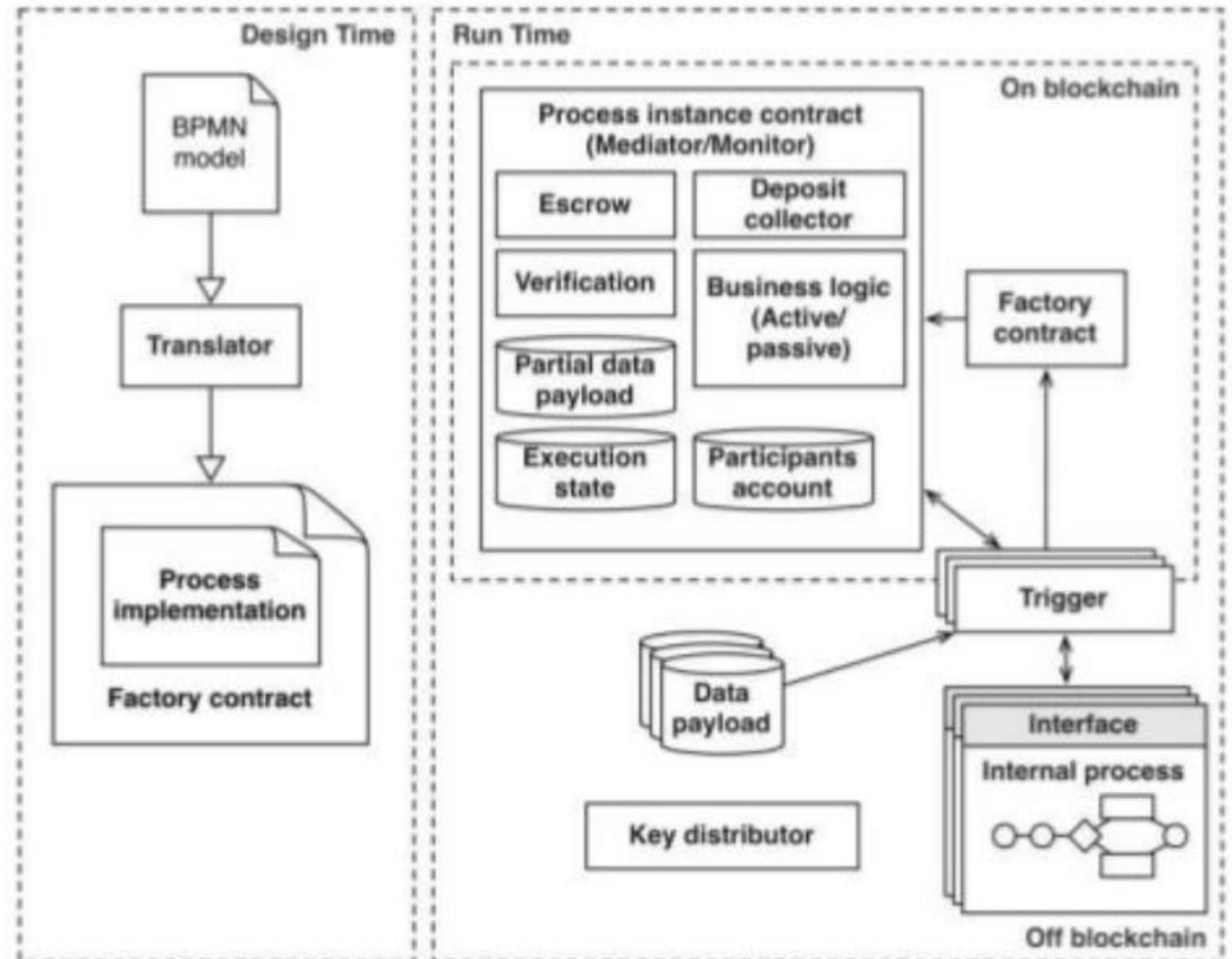


High Level Architecture

Business Process Monitoring and Execution

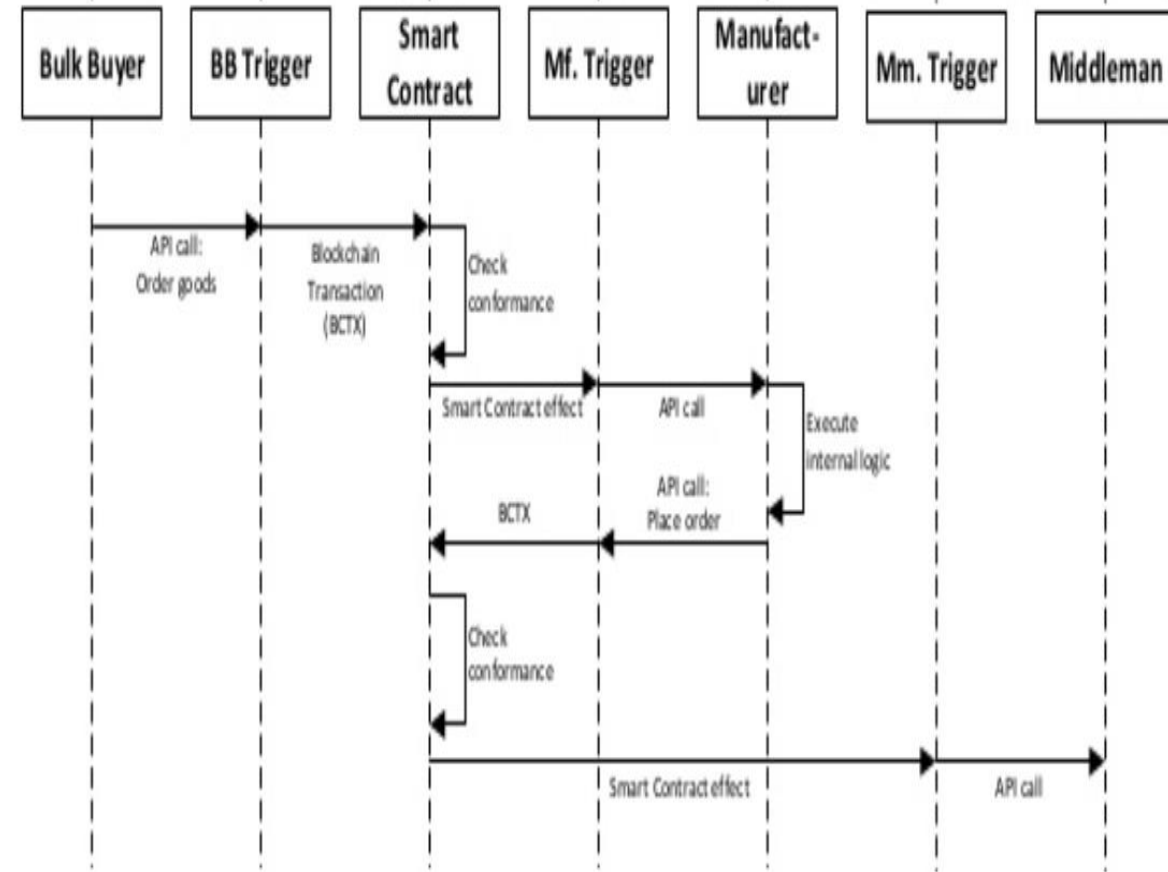
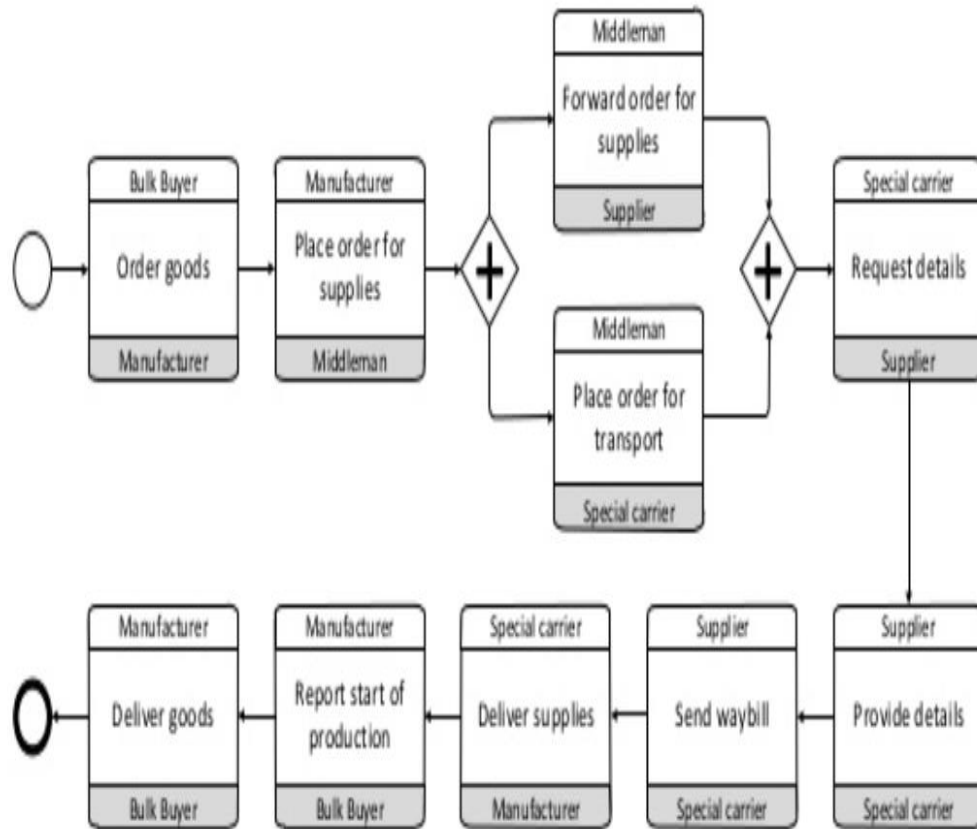
Approach

- Translate Business Process Model and Notation (BPMN) to smart contract code.
- Triggers act as bridge between Enterprise world and Blockchain.
- Smart Contract does the following:
- Independent, global process monitoring
 - Conformance checking
 - Automatic payment and ticketing
 - Data transformation
 - Automated rewards program
 - Encryption



High Level Architecture

Business Process Monitoring and Execution- Translation & Instantiation



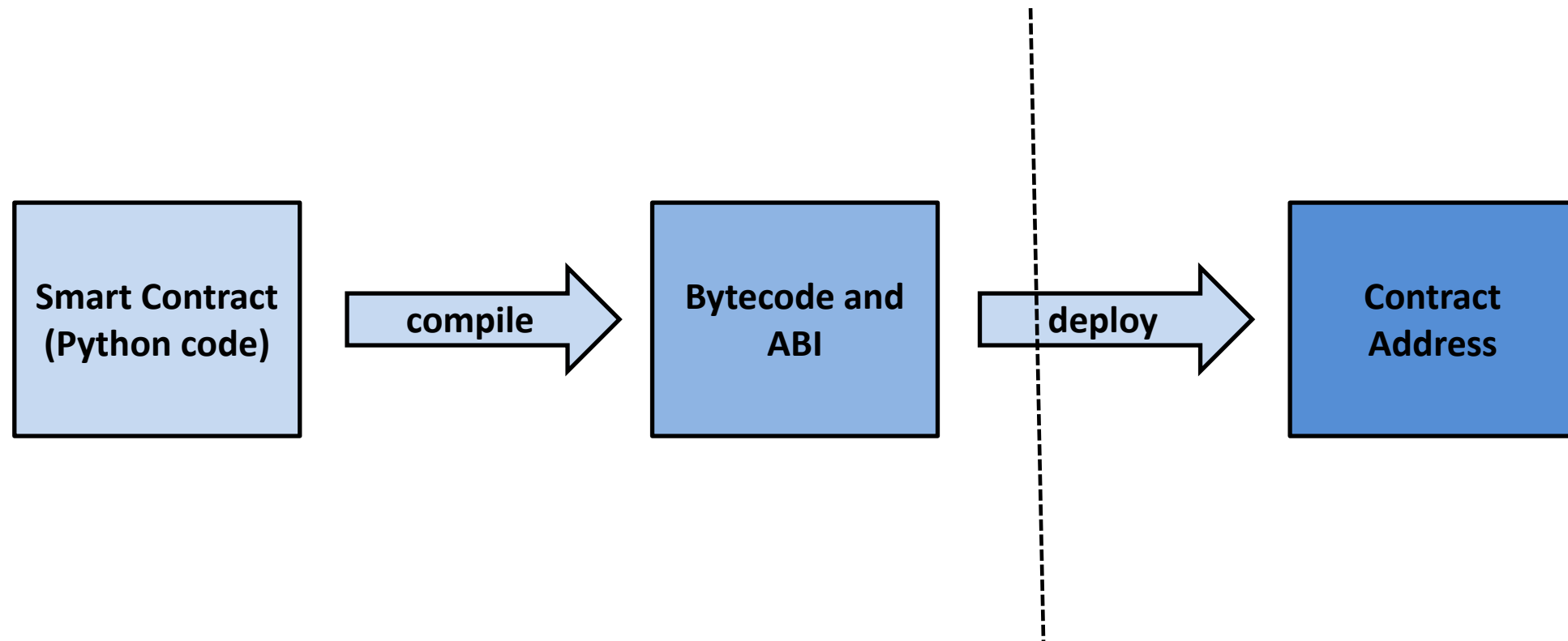
Development Environments

Environment	NEO	Ethereum	Pythereum
Access Method	NEO-BOA, JSON RPC API, NEX ICO, NEO Smart Contract	Remix/Node Console (NodeJS,) & Web3/Mist & Ethereum Wallet	Remix/Node Console & Web3/Mist & Ethereum Wallet
Blockchain Technology	NEO Virtual Machines	JavaScript VM inside Remix/TestRPC/Private Ethereum Blockchain	JavaScript VM inside Remix/TestRPC/Private Ethereum Blockchain
User Interface and Development Language	Neo-Python, AngularJS/ReactJS, APIs (Django, Flask, etc.)	Solidity, AngularJS/ReactJS, APIs (Django, Flask, etc.)	Python, AngularJS/ReactJS, APIs (Django, Flask, etc.)
Comments	<ul style="list-style-type: none"> • Good Community Support for Developers (City of Zion). • Easy implementation of Machine Learning/ Artificial Intelligence based solutions. 	<ul style="list-style-type: none"> • Most popular tool to implement Blockchain based solutions. • Very strong community support. 	<ul style="list-style-type: none"> • Can be used to explore Ethereum functionalities in Python. • Easy implementation of Machine Learning/ Artificial Intelligence based solutions.

Development Environments

NEO Blockchain

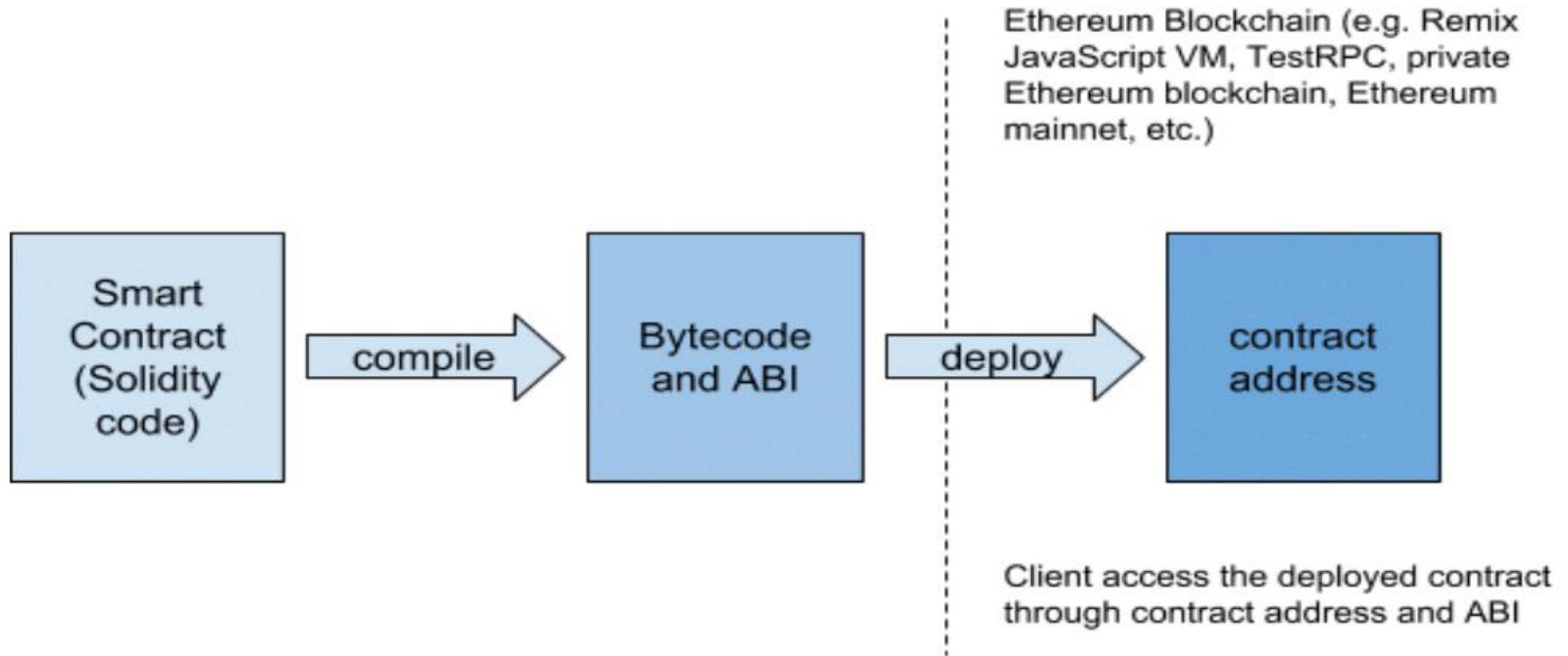
A Simplified flow of Contract Compilation and Deployment



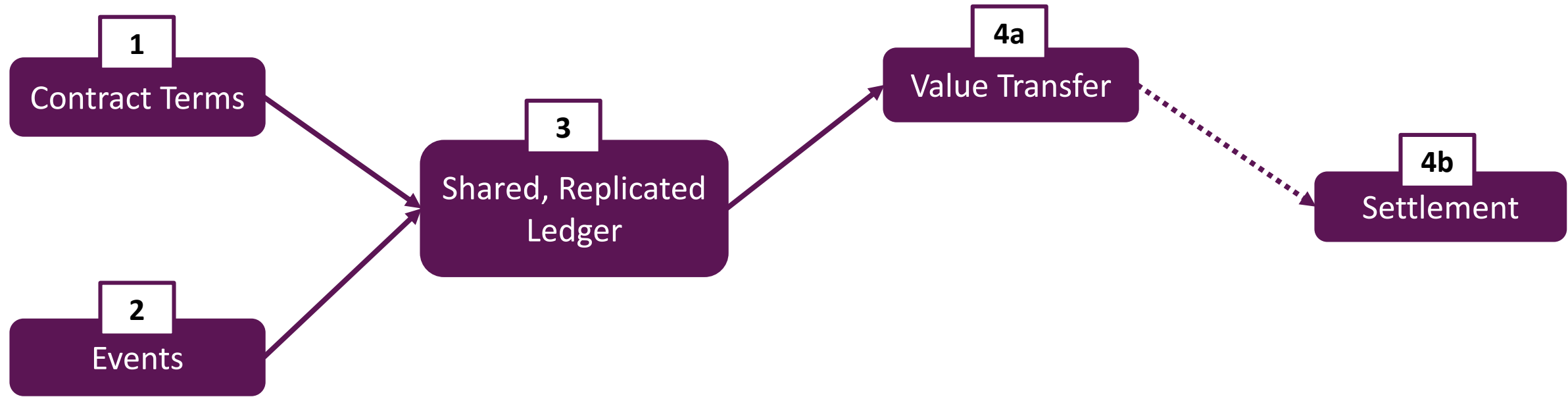
Development Environments

Ethereum/Pythereum Blockchain

A Simplified Flow of Contract Compilation and Deployment



Applying Business Logic with Smart Contracts



1

- Counterparties establish obligations.
- Assets put under custody of smart contract.
- Conditions for execution.

2

- Event triggers contract execution.
- Event can refer to:
 - Transaction Initiation
 - Information Received

3

- Terms of Contract dictate movement of value based on conditions met.

4a

- Value Transferred to intended recipient as dictated by contract terms.
- For digital Assets on-chain (e.g. Bitcoin, Digital Money), accounts are automatically settled.

4b

- For assets represented off-chain (e.g. Securities, Cases, Real Estate), value could be moved and settled in off-chain accounts per settlement instructions.

Proof of Concepts

Case 01

- Objective: Smart Contracts for Case Processing within an organization.
- Environment: Ethereum
- Development Language: Solidity and JavaScript.
- Nodes are various departments/segments/groups within an organization.
- Nodes interact with each other for transactions related to some processes.

Sample Code

Case 02

- Objective: Smart Contracts for Case Processing within an organization.
- Environment: Ethereum
- Development Language: Solidity and JavaScript.
- Nodes are various departments/segments/groups within an organization.
- Nodes interact with each other for transactions related to some processes.

Sample Code

Please Note:

The PoC/Cases are simple processes and included for demonstration purpose only. Will share a PoC relevant to the SESS process in 2 months' time. Currently, spending time to get a high level understanding of various sub-processes.

Thank You