



Iroha DSL

Nikita Puzankov

Senior Rust Dev
puzankov@soramitsu
.co.jp

Agenda

- . Review of the Iroha Special Instructions concept
- . Comparison with Commands in Iroha 1
- . RFC about Instructions DSL
- . Prototype overview
- . Q&A

Review of the Iroha Special Instructions concept

Background

Smart contracts have the potential to automate business processes and create logic that can be executed without human interaction

More details in [White Paper](#)

Requirements

- . Turing complete logic execution
 - transactions
 - triggers

Requirements

- . Turing complete logic execution
 - assets, accounts and domains management
 - peers configuration
 - mathematics

Comparison with Commands in Iroha 1

Background

A command is an intention to change the state of the network. For example, in order to create a new role in Iroha you have to issue Create role command.

More details in [Read the Docs](#)

Create Domain example - Iroha 1

The purpose of create domain command is to make new domain in Iroha network, which is a group of accounts.

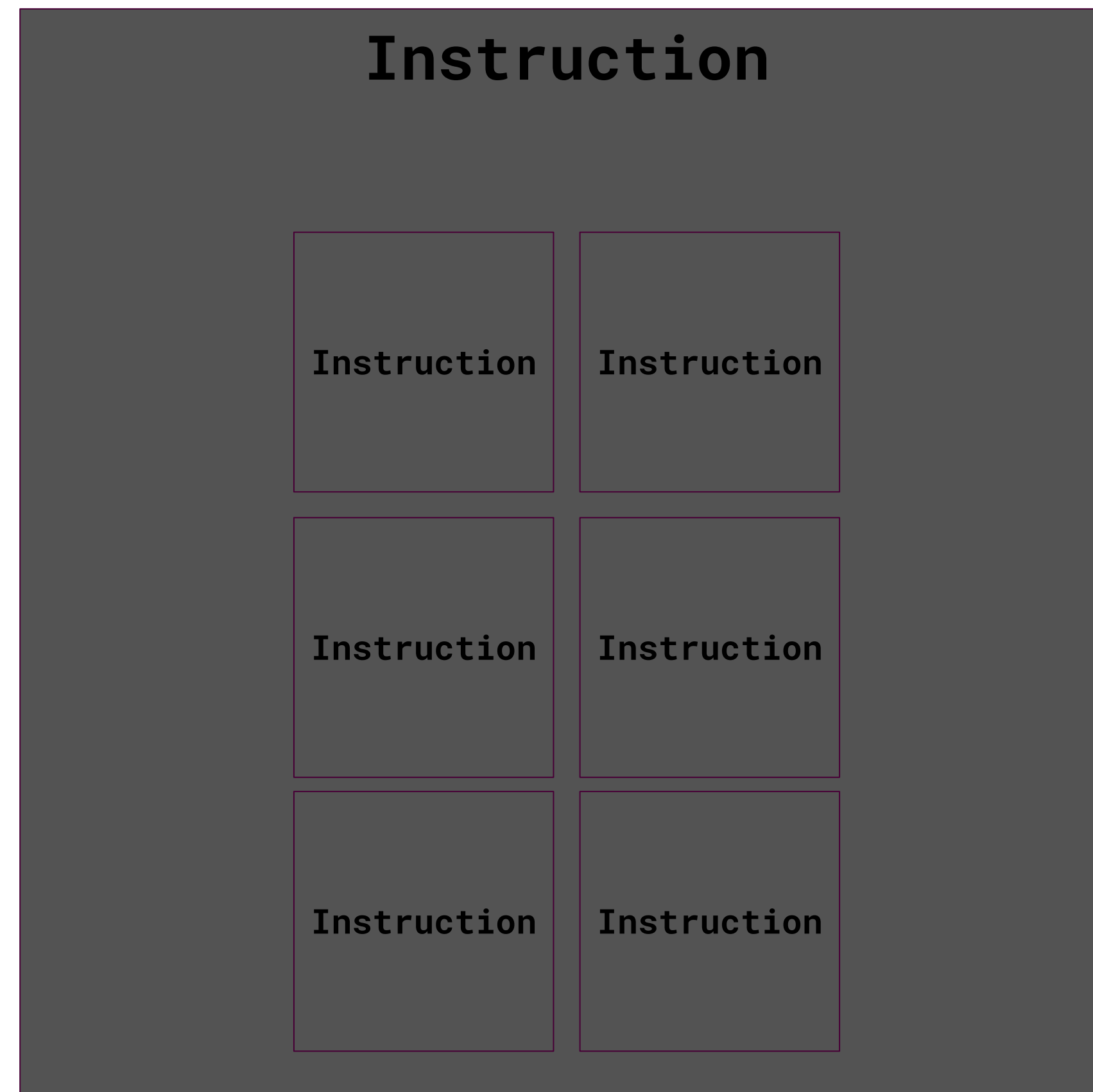
```
/**
 * Create new asset in the system
 */
struct CreateDomain : public Command {
    /**
     * Asset to insert to the system
     */
    std::string domain_id;

    /**
     * Default role for users in the domain
     */
    std::string user_default_role;
```

Create Domain example - Iroha 2

```
50 /// Generic instruction for a registration of an object to the identifiable
destination.
51 #[derive(Debug)]
52 pub struct Register<D, O>
53 where
54     D: Identifiable,
55 {
56     /// Object which should be registered.
57     pub object: O,
58     /// Destination object `Id`.
59     pub destination_id: D::Id,
60 }
...
160 impl Instruction for Register<Peer, Domain> {}
```


Composition of Instructions



RFC about Instructions DSL

Solution

- Out-of-the-box set of “possible” instructions
- An ability to combine them to create custom instructions
- Iroha Triggers to execute instructions:
 - On time
 - On World State View changes
 - On Blockchain changes

Domain entities management

- . Register/Unregister (account or asset definition in a domain)
- . **Add/Remove** (signatory to or from an account, trigger to the peer, domain to the peer, trusted peer to the peer)
- . Transfer (asset from one account to another)
- . Mint/Demint (asset quantity)

Triggers

- . OnBlockCreated
- . OnBlockchainHeight
- . OnWorldStateViewChange
- . OnTimestamp

Mathematics

- . Add
- . Subtract
- . Multiply
- . Divide
- . RaiseTo
- . Mod

Permissions

- . Anything
- . AddDomain
- . RemoveDomain
- . AddTrigger
- . RemoveTrigger
- . RegisterAssetDefinition

...

Queries

Execute(IrohaQuery)

Composable types

- . Pair(Instruction1, Instruction2)
- . Sequence(Vec<Instruction>)
- . If(Condition<Instruction>, Then<Instruction>, Else<Instruction>)

Prototype Overview

Example

FindRateAndCheckItGreaterThanValue

DSL - Instructions Composition

```
62     let _ = If::new(  
63         Box::new(Not::new(Box::new(Greater::new(  
64             FindAssetQuantityById::new(AssetId::from_names(  
65                 "btc2eth_rate",  
66                 "exchange",  
67                 "dex",  
68                 "exchange",  
69             )),  
70             10,  
71         )))),  
72         Box::new(Fail::new("rate is less or equal to value")),  
73     );
```


Custom Instruction Definition

```
76 struct FindRateAndCheckItGreaterThanValue {
77     from_currency: String,
78     to_currency: String,
79     value: u32,
80 }
81
82 impl FindRateAndCheckItGreaterThanValue {
83     pub fn new(from_currency: &str, to_currency: &str, value: u32) -> Self {
84         FindRateAndCheckItGreaterThanValue {
85             from_currency: from_currency.to_string(),
86             to_currency: to_currency.to_string(),
87             value,
88         }
89     }
}
```

Custom Instruction Definition

```
...
90
91     pub fn into_isi(self) -> If {
92         If::new(
93             Box::new(Not::new(Box::new(Greater::new(
94                 FindAssetQuantityById::new(AssetId::from_names(
95                     &format!("{}", self.from_currency, self.to_currency),
96                     "exchange",
97                     "dex",
98                     "exchange",
99                 )),
100             self.value,
101             )))),
102         Box::new(Fail::new("rate is less or equal to value")),
103     )
104 }
105 }
```

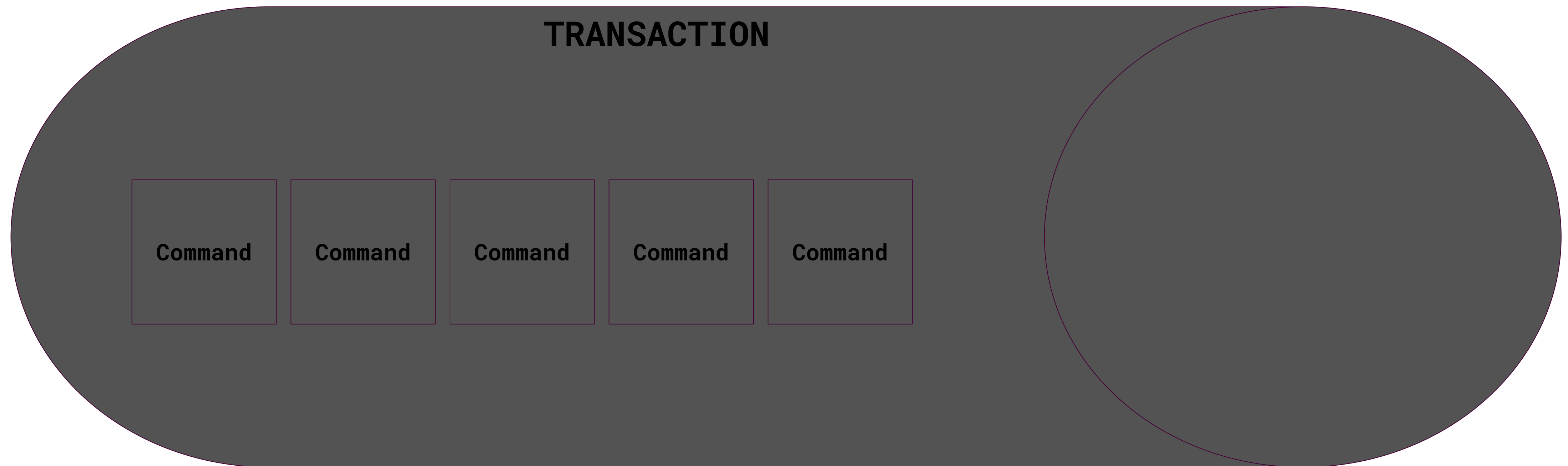
Q&A

Composition

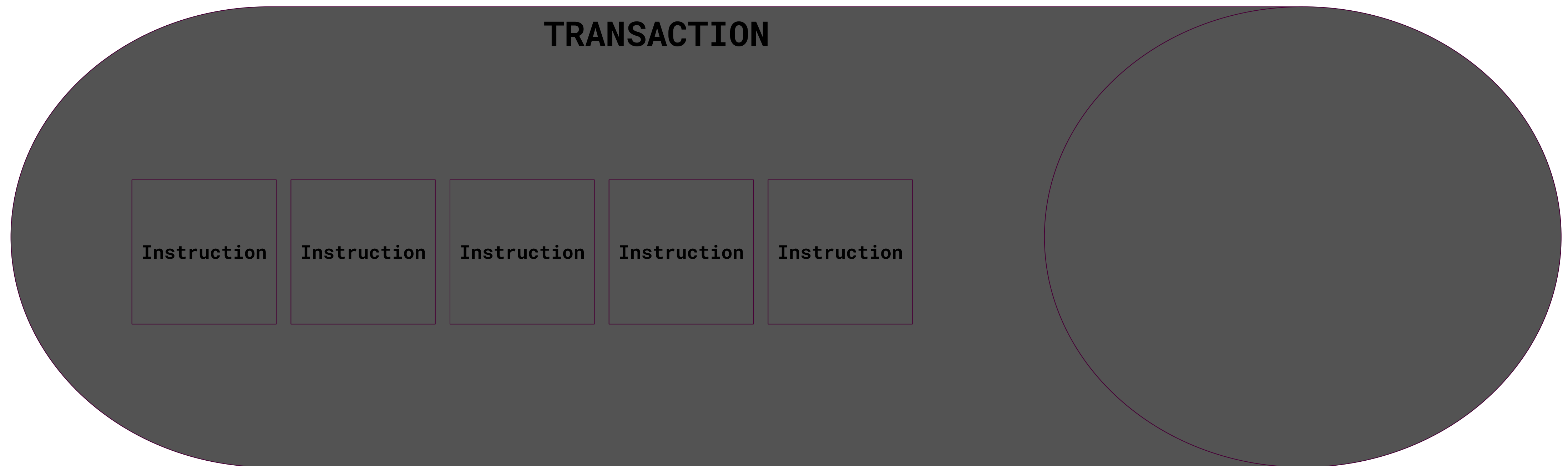
Command \neq Transaction

Instruction \neq Transaction

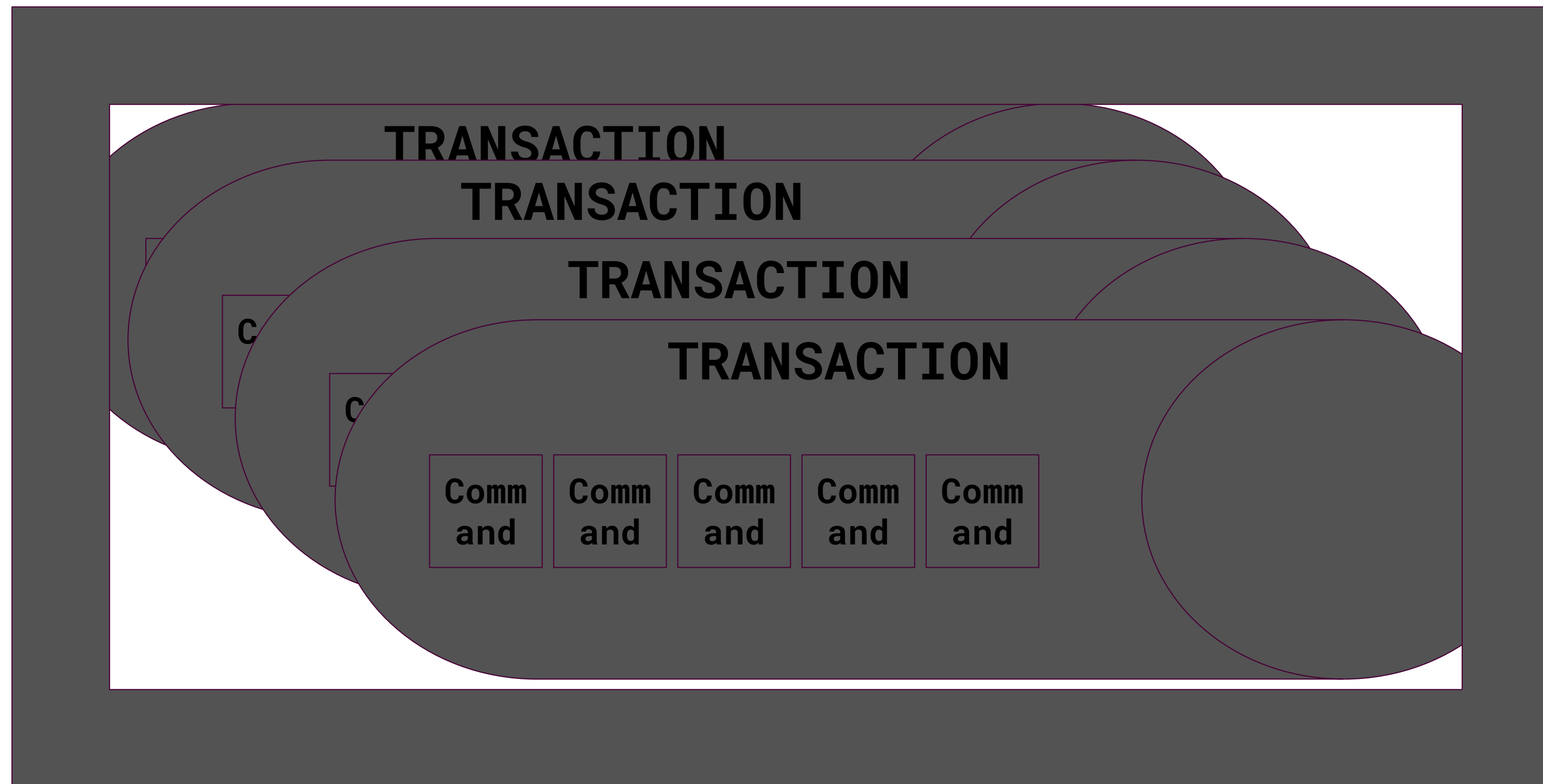
Composition - Transaction



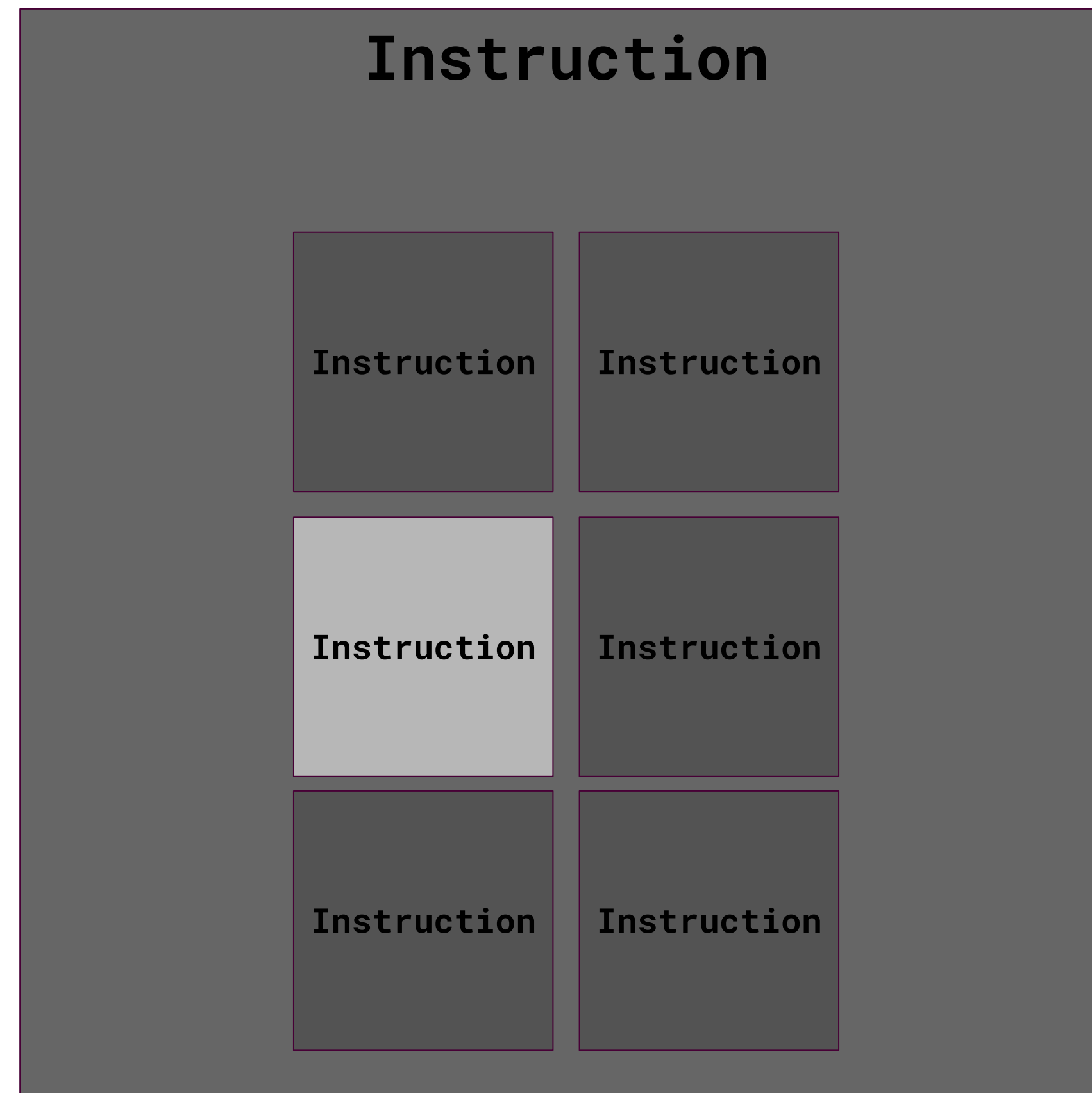
Composition - Transaction



Composition - Transactions Batch



Composition of Instructions



Permissions

- . Permissions check - part of Instruction impl
- . Custom permissions checks effective only inside Triggers

Triggers

Events Listeners concept described in White Paper.
Additional design decisions will be made in RFC.

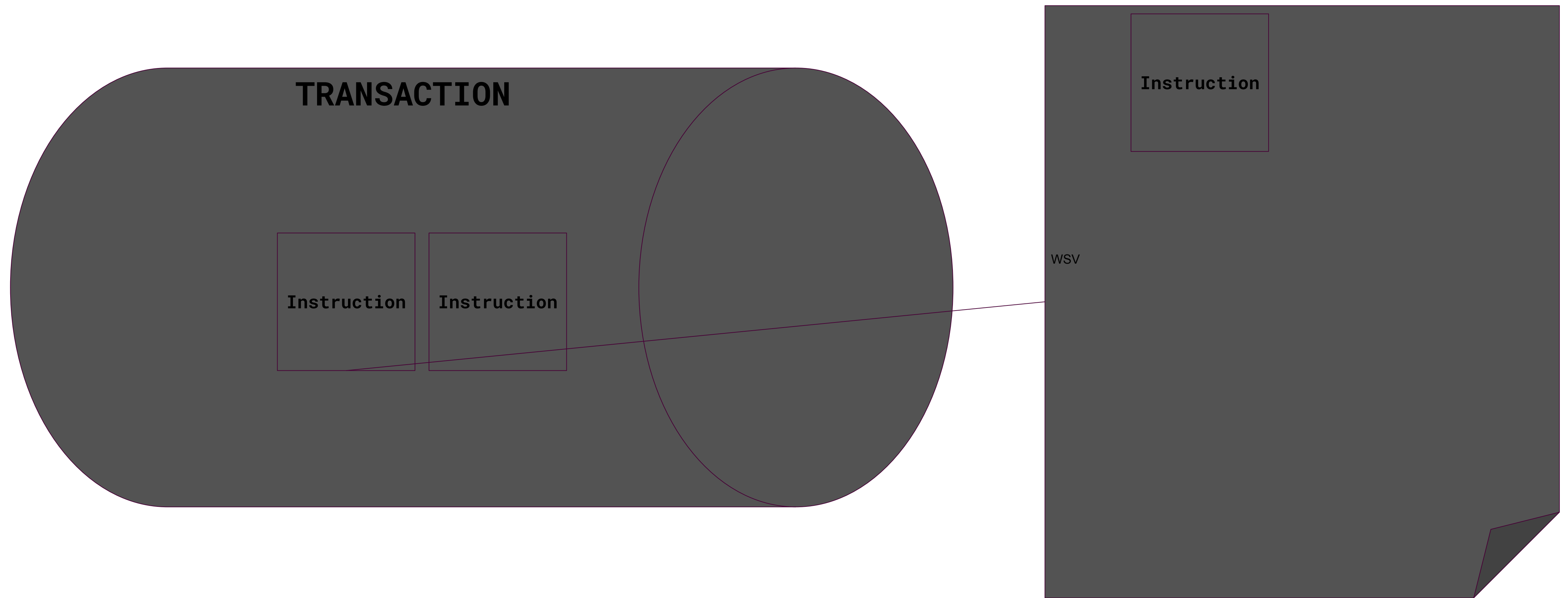


Questions?

Nikita Puzankov

Senior Rust
Developer
puzankov@soramitsu
.co.jp

Composition - Transaction



Public API

Client - ISI
Java → Json
C++ → Json
...

Peer - ISI
Json → Rust

WSV

Data Model

- Python Adapter
- Transaction Execution
 - Blocks Management
 - ...

Public API

Blockchain