



# Iroha 2 - Expressions for ISI

Egor Ivkov

---

Rust Developer  
ivkov@soramitsu.co.jp

# ISI - Iroha Special Instructions

---

- Execute logic on ledger
- Smart Contract Language for Iroha
- Transactions are composed of ISIs
- Triggers are composed of ISIs
- Common ISI: Mint, Burn, Transfer, etc.
- Special ISI: Pair, Sequence, If, etc.

# Limitations of ISI

```
28 implementations
/// Trait implementations should provide actions to apply changes on `WorldStateView`.
pub trait Execute {
    /// Apply actions to `world_state_view` on behalf of `authority`.
    fn execute(
        self,
        authority: <Account as Identifiable>::Id,
        world_state_view: &WorldStateView,
    ) -> Result<WorldStateView, String>;
}
```

- ISI implement **Execute** trait.
- **Execute** trait let's an instruction to return a mutated **WSV** or an **Error**.
- ISI are always mutable - there was no possibility to perform a calculation without mutating the state.

# Expressions

```
15 implementations
// Calculate the result of the expression without mutating the state.
pub trait Evaluate {
    // The resulting type of the expression.
    type Value;

    // Calculates result.
    fn evaluate(
        &self,
        world_state_view: &WorldStateView,
        context: &Context,
    ) → Result<Self::Value, String>;
}
```

- Take immutable reference to **WSV**
- Return a **Value**
- Common Expressions: Add, Subtract, Greater, Less, Equal, etc.
- Special Expressions: If, Where, ContextValue, etc.

# Use Case - Whitepaper Example

---

Conditional Multisignature

From whitepaper:

```
Condition.asset("usd@nbc").qty(500).comparison(">").qty(1000).comparison("<")
```

# Use Case - Implementation

```
let condition: ExpressionBox = IfBuilder::condition(And::new(  
  left: Greater::new(left: ContextValue::new(value_name: "usd_quantity"), right: 500u32),  
  right: Less::new(left: ContextValue::new(value_name: "usd_quantity"), right: 1000u32),  
)): IfBuilder  
.then_expression(Or::new(  
  left: ContainsAll::new(  
    collection: ContextValue::new(value_name: "signatories"),  
    elements: teller_signatory_set.clone(),  
  ),  
  right: Contains::new(collection: ContextValue::new(value_name: "signatories"), element: manager_signatory),  
)): IfBuilder  
.else_expression(true): IfBuilder  
.build()?: If  
.into();
```

# Expressions in Instructions

```
3 implementations
/// Sized structure for all possible Transfers.
#[derive(Debug, Clone, Serialize, Deserialize, Encode, Decode)]
pub struct TransferBox {
    /// Entity to transfer from.
    pub source_id: EvaluatesTo<IdBox>,
    /// Object to transfer.
    pub object: EvaluatesTo<Value>,
    /// Entity to transfer to.
    pub destination_id: EvaluatesTo<IdBox>,
}
```

**EvaluatesTo<V>** is a typecheck structure holding an expression.

Raw values are also considered to be expressions.



# Questions?

Egor Ivkov

Rust Developer  
[ivkov@soramitsu.co.jp](mailto:ivkov@soramitsu.co.jp)

---