

# Java SCORE Overview

v0.6

June 2022

# Why Java and the JVM?

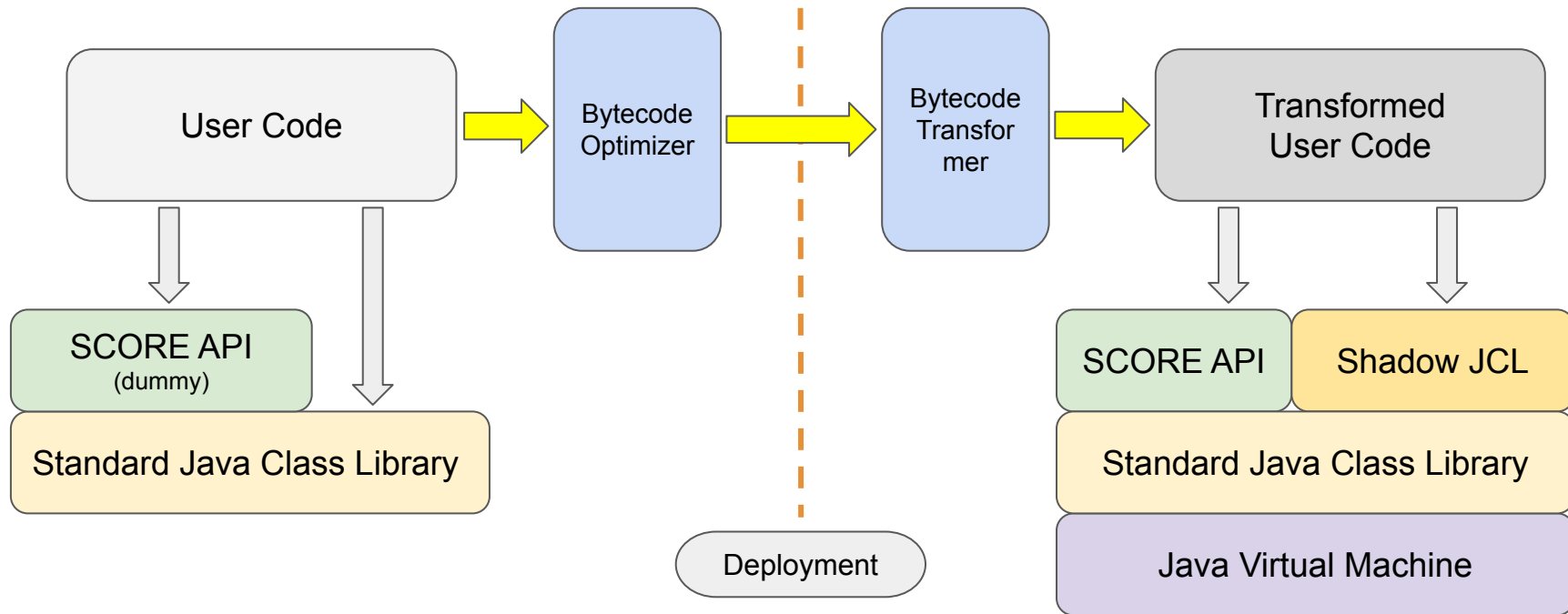
## Pros

- Reliable and mature implementation
- Strong type system
- Extensive Java tooling support
- JIT performance
- Java language popularity
- **Sandbox, Sandbox, Sandbox!**

## Cons

- Non-deterministic operation inherently

# Bytecode Instrumentation



# Bytecode Instrumentation

## Bytecode Optimizer

- Extracts all annotations (`@External`, ...) and builds a APIs mapping
- Generates the body of methods tagged with `@EventLog`
- Removes unreachable methods
- Renames all the class, method, and field names to smaller names (a.k.a. obfuscation) to reduce the deployment cost (*optional*)

## Bytecode Transformer

- Checks if bytecode uses only APIs which are in whitelist
- Remaps reference to JCL classes into shadow classes
- Inserts code for charging instruction cost
- Removes '`<clinit>`', and many other tasks...

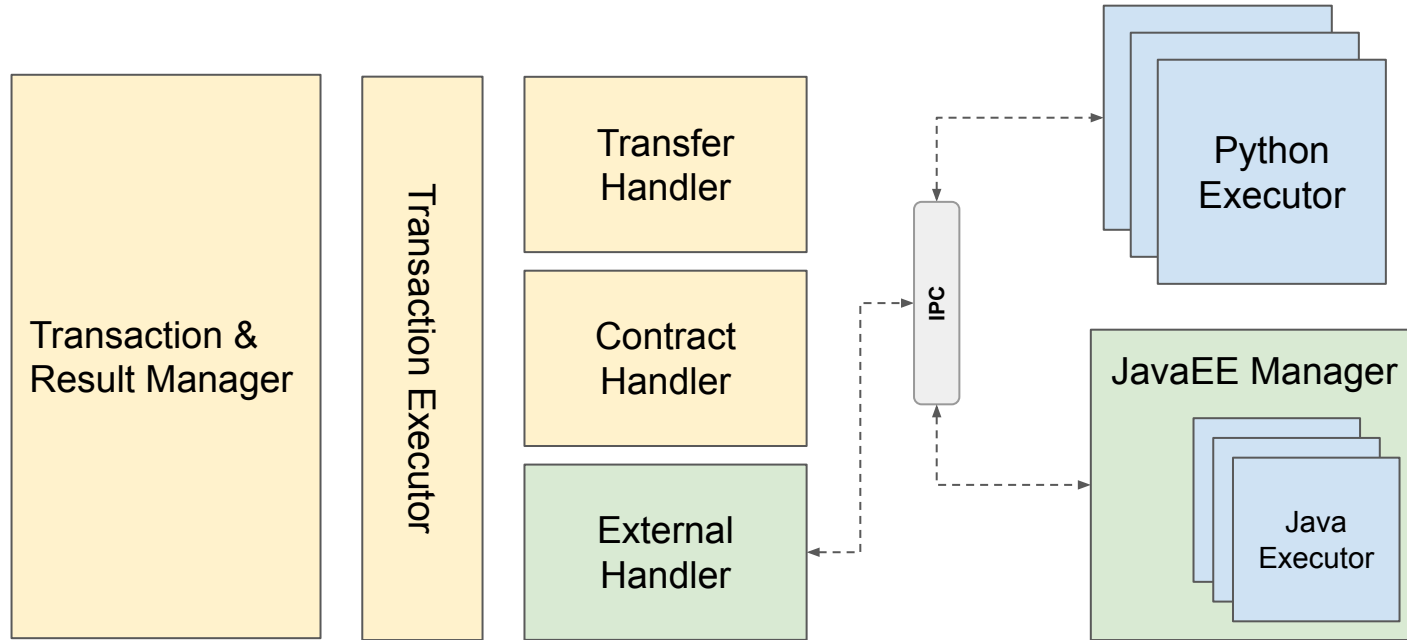
# SCORE Structure Comparison

| Name               | Python                                       | Java                                     |
|--------------------|----------------------------------------------|------------------------------------------|
| External decorator | <code>@external</code>                       | <code>@External</code>                   |
|                    | <code>@external(readonly=True)</code>        | <code>@External(readonly=true)</code>    |
| Payable decorator  | <code>@payable</code>                        | <code>@Payable</code>                    |
| Eventlog decorator | <code>@eventlog</code>                       | <code>@EventLog</code>                   |
|                    | <code>@eventlog(indexed=1)</code>            | <code>@EventLog(indexed=1)</code>        |
| fallback signature | <code>def fallback(self)</code>              | <code>public void fallback()</code>      |
| SCORE initialize   | override <code>on_install(...)</code> method | define a <code>public constructor</code> |
| Default parameters | native language support                      | <code>@Optional</code>                   |

# SCORE APIs Comparison

| Python                                 | Java                                      | Description                                                      |
|----------------------------------------|-------------------------------------------|------------------------------------------------------------------|
| <code>self.address</code>              | <code>Context.getAddress()</code>         | The current SCORE address                                        |
| <code>self.block.height</code>         | <code>Context.getBlockHeight()</code>     | The current block height                                         |
| <code>self.msg.sender</code>           | <code>Context.getCaller()</code>          | The address of the account who called this function              |
| <code>self.tx.hash</code>              | <code>Context.getTransactionHash()</code> | The transaction hash                                             |
| <code>self.icx.transfer()</code>       | <code>Context.transfer()</code>           | ICX transfer function                                            |
| <code>revert()</code>                  | <code>Context.revert()</code>             | Reverts the transaction and breaks                               |
| <code>create_address_with_key()</code> | <code>Context.getAddressFromKey()</code>  | Returns the address that is associated with the given public key |

# Component Diagram



# Invoking a external method of other SCOREs

- Use *Context.call()*

```
// [package score.Context]
public static Object call(Address targetAddress, String method,
                          Object... params);

public static Object call(BigInteger value, Address targetAddress,
                          String method, Object... params);

// Example
if (_to.isContract()) {
    Context.call(_to, "tokenFallback", _from, _value, dataBytes);
}
```

# Error Handling

- Uncatchable exceptions
  - Non-deterministic system errors (eg. OOM or I/O errors)
    - shall not be visible to SCORE (uncatchable), otherwise SCORE's behavior may depend on the error.
  - Deterministic, but shall not be catchable
    - OutOfStep, StackOverflow
- Catchable exceptions
  - IllegalArgumentException
    - {Contract|Method}NotFound, OutOfBalance, InvalidParameter, etc
  - ScoreRevertException
    - Caused by *Context.revert()*

# Support List/Map return types in @External method

- Provide *List.of()* and *Map.of()*
  - Convenience static factory methods for creating *unmodifiable* instances
  - [JEP 269: Convenience Factory Methods for Collections](#)
  - Fully compliant with the existing Python SCOREs

```
@External(readonly=true)
public List<Address> getConfirmations(BigInteger _transactionId) {
    ...
    Address[] confirmations = new Address[count];
    ...
    return List.of(confirmations);
}
```

# Storage Model

## Object Graph

- Serializes all live objects that are reachable from static roots
- Provides an illusion like this:
  - ***A program starts and then runs forever.***
- Cons: inefficient as it must be stored as a whole even if it is partially changed

## Key-Value Storage

- Supports same interfaces as Python SCORE does
  - VarDB, ArrayDB and DictDB

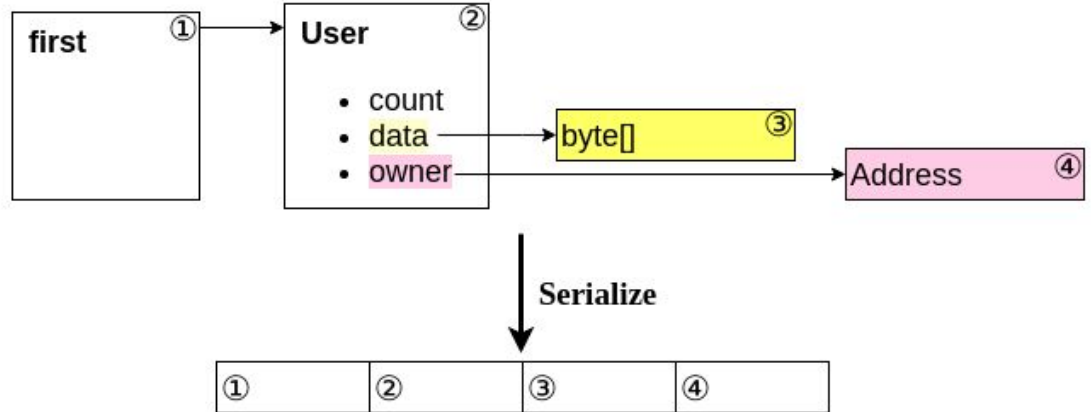
# Object Graph

## Code

```
class User {  
    int count;  
    byte[] data;  
    Address owner;  
}  
  
static User first;
```

=

## Automatic Object Graph



# Variable DB

```
// VarDB<BigInteger> counter = ...  
var counter = Context.newVarDB("counter", BigInteger.class);  
  
counter.set(BigInteger.ZERO);  
counter.set(counter.get().add(BigInteger.ONE));
```

# Array DB

```
// ArrayDB<Address> addrList = ...  
var addrList = Context.newArrayDB("addrList", Address.class);  
  
addrList.add(Context.getAddress());  
Context.require(addrList.size() == 1);  
Context.require(addrList.get(0).equals(Context.getAddress()));  
  
addrList.set(0, Context.getOwner());  
Context.require(addrList.get(0).equals(Context.getOwner()));
```

# Dictionary DB

```
DictDB<Address, BigInteger> balances  
    = Context.newDictDB("balances", BigInteger.class);  
  
var balance = BigInteger.valueOf(1_000_000);  
balances.set(Context.getOwner(), balance);  
Context.require(balances.get(Context.getOwner()).equals(balance));
```

# Branch DB

- A branch DB behaves as branches to sub-DBs

```
BranchDB<BigInteger, DictDB<Address, Boolean>> confirmations
    = Context.newBranchDB("confirmations", Boolean.class);

var txID = BigInteger.ZERO;
confirmations.at(txID).set(Context.getCaller(), true);
Context.require(confirmations.at(txID).get(Context.getCaller()));
```

# Object Serialization in DB (1)

- A DB can store an object of a *CustomClass*, if the class implements the following codec methods.

```
public static void writeObject(ObjectWriter w, CustomClass v) {  
    ...  
}  
  
public static CustomClass readObject(ObjectReader r) {  
    ...  
}
```

# Object Serialization in DB (2)

```
public class Transaction {  
    public static void writeObject(ObjectWriter w, Transaction t) {  
        w.writeListOf(t.from, t.to, t.value);  
    }  
  
    public static Transaction readObject(ObjectReader r) {  
        r.beginList();  
        Transaction t = new Transaction(  
            r.read(Address.class),  
            r.read(Address.class),  
            r.read(BigInteger.class)  
        );  
        r.end();  
        return t;  
    }  
}
```

# Object Serialization in DB (3)

```
DictDB<BigInteger, Transaction> transactions  
    = Context.newDictDB("transactions", Transaction.class);
```

```
var tx1 = new Transaction(  
    Context.getOrigin(),  
    Context.getOwner(),  
    BigInteger.valueOf(1_000_000)  
);  
transactions.set(BigInteger.valueOf(1), tx1);
```

```
var tx2 = transactions.get(BigInteger.valueOf(1));  
Context.require(tx1.equals(tx2));
```

# References

- SCORE API document
  - <https://www.javadoc.io/doc/foundation.icon/javaee-api>
- gradle-javaee-plugin project
  - <https://github.com/icon-project/gradle-javaee-plugin>
- Java SCORE examples project
  - <https://github.com/icon-project/java-score-examples>
- javaee project
  - <https://github.com/icon-project/goloop/tree/master/javaee>
- goloop CLI commands
  - [https://github.com/icon-project/goloop/blob/master/doc/goloop\\_cli.md](https://github.com/icon-project/goloop/blob/master/doc/goloop_cli.md)