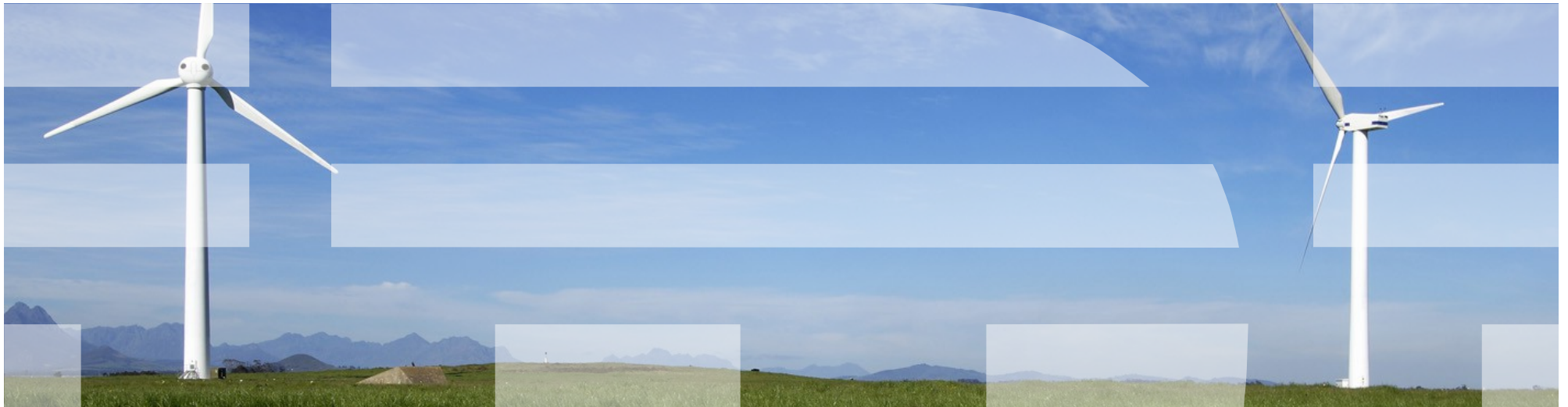
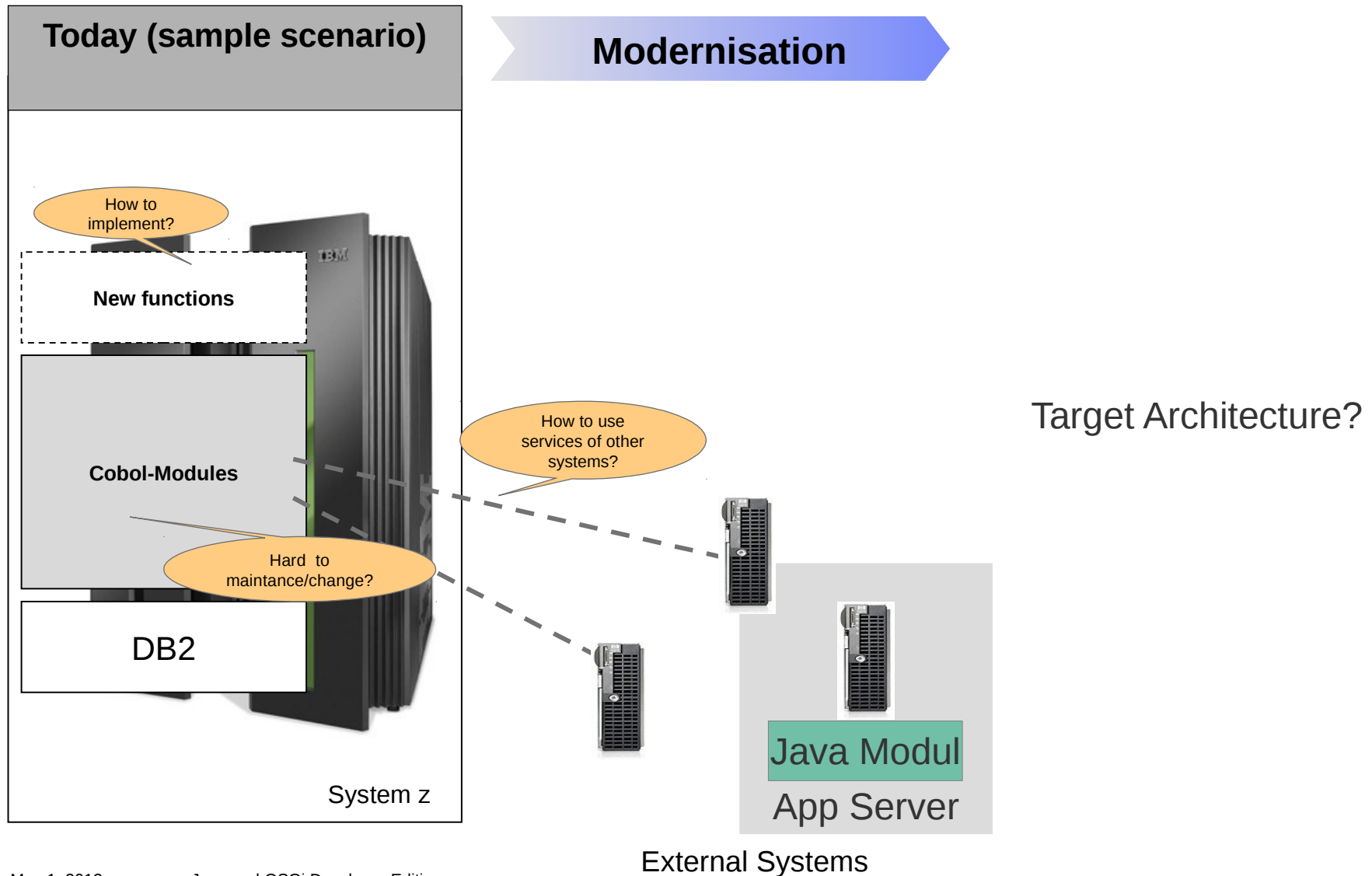


Java – Yet another Language

on the mainframe



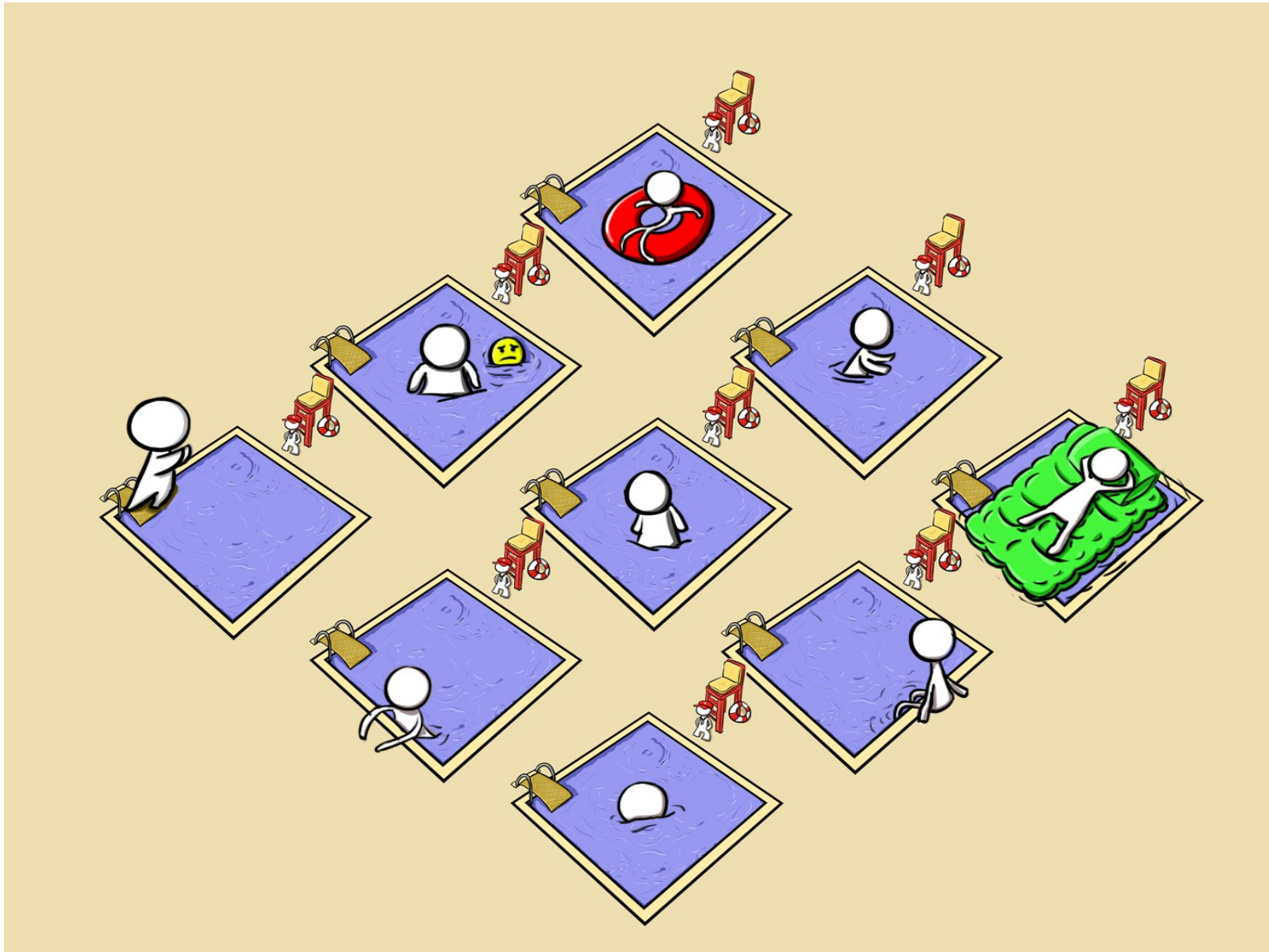
Big Picture: Where to go with the CICS Development



A Java Anology



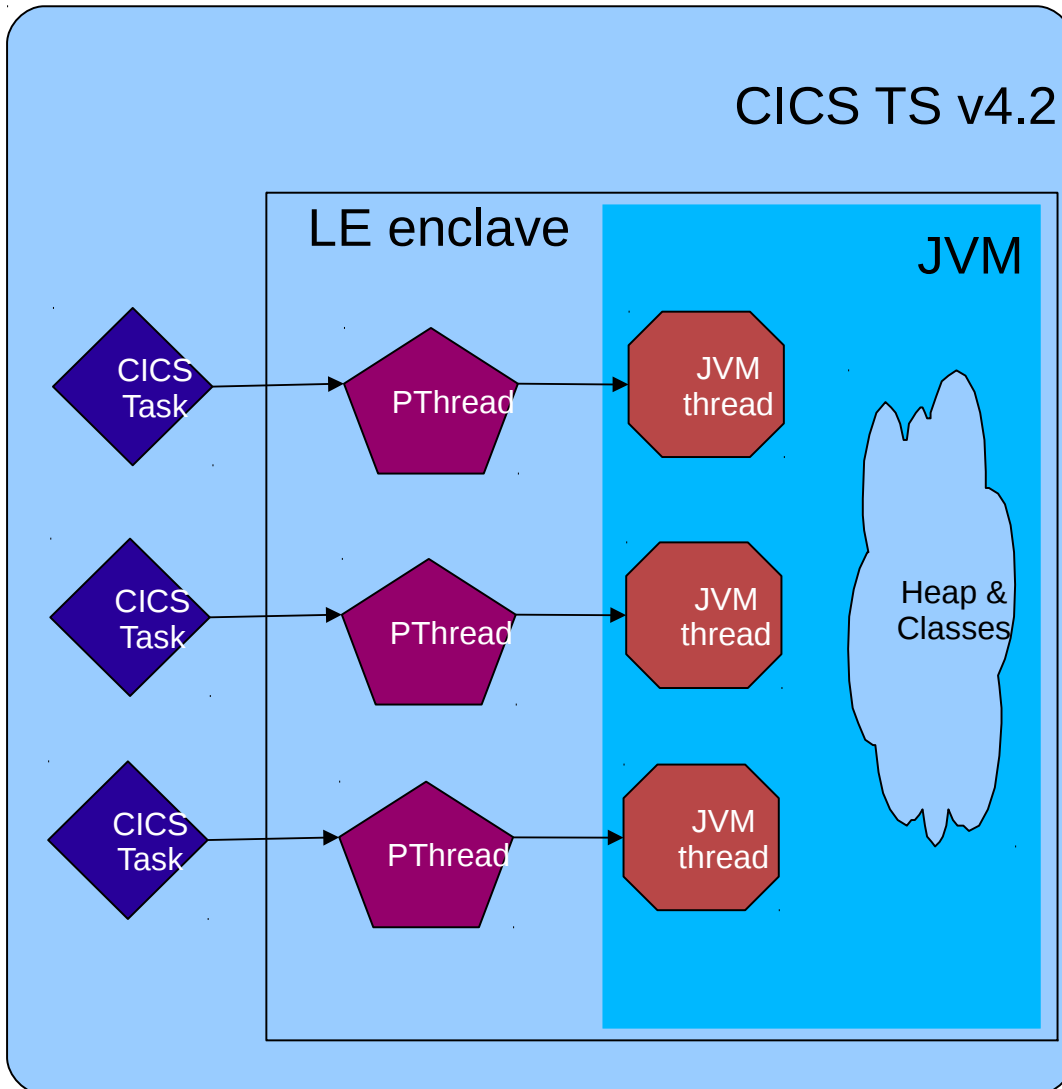
Flashback: CICS Pooled Server Architecture



The JVM Server



JVM servers



- CICS requests storage from MVS, sets up a Language Environment enclave, and launches the 64-bit JVM in the enclave.
- IBM® 64-bit SDK for z/OS, Java Technology Edition, Version 6.0.1
- Up to 256 parallel tasks/JVM & 1024/CICS
- Applications
 - Must be threadsafe
 - deployed as OSGi bundles (in CICS bundles)
- Dynamic updates without restart
- No EJB support

***Java* is just another Language which is running on the Mainframe**

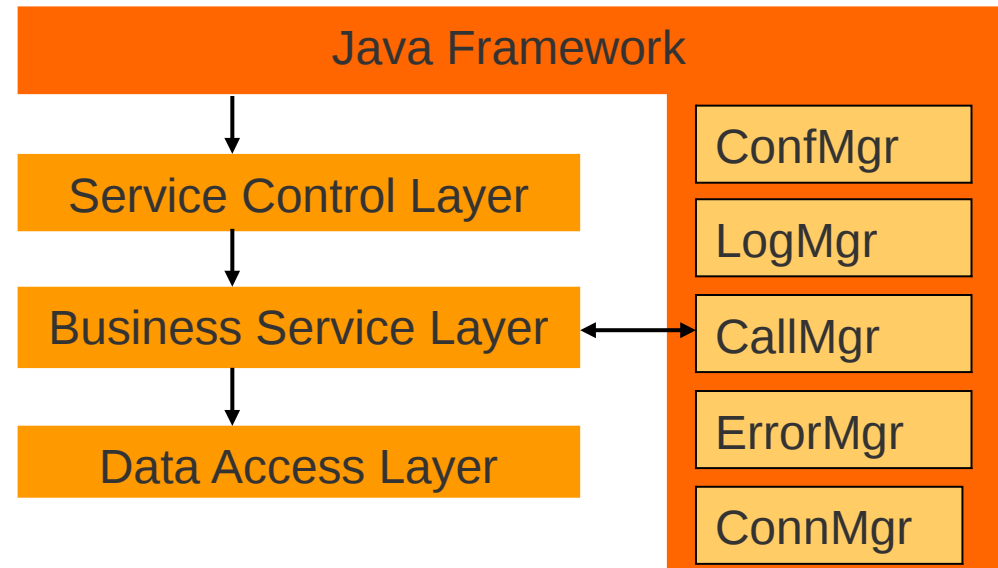
Which tools did we discovered yet?



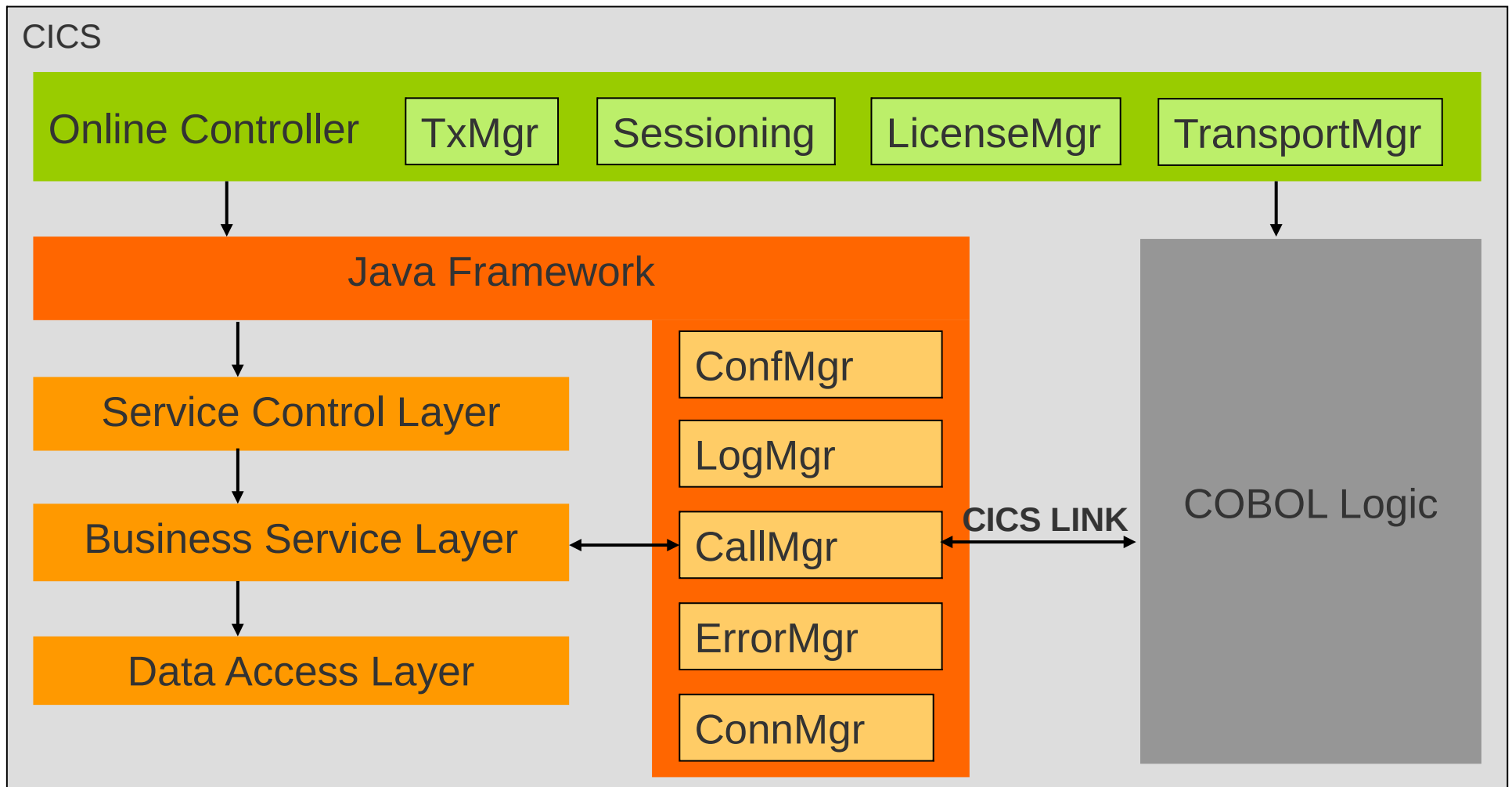
- Java:
 - Language that is able to run on the host
 - Object oriented approach
- OSGi:
 - Service Oriented Approach
 - Exchange of modules that are in Service
 - Available in nearly all Java environments
- So what is missing?
 - » How is data accessed?
 - » How are connections managed?
 - » How is the output formatted?
 - » How can other programs be integrated?
- The following approach was developed by our service department led by Philipp Breitbach

The application needs to be designed within a Framework

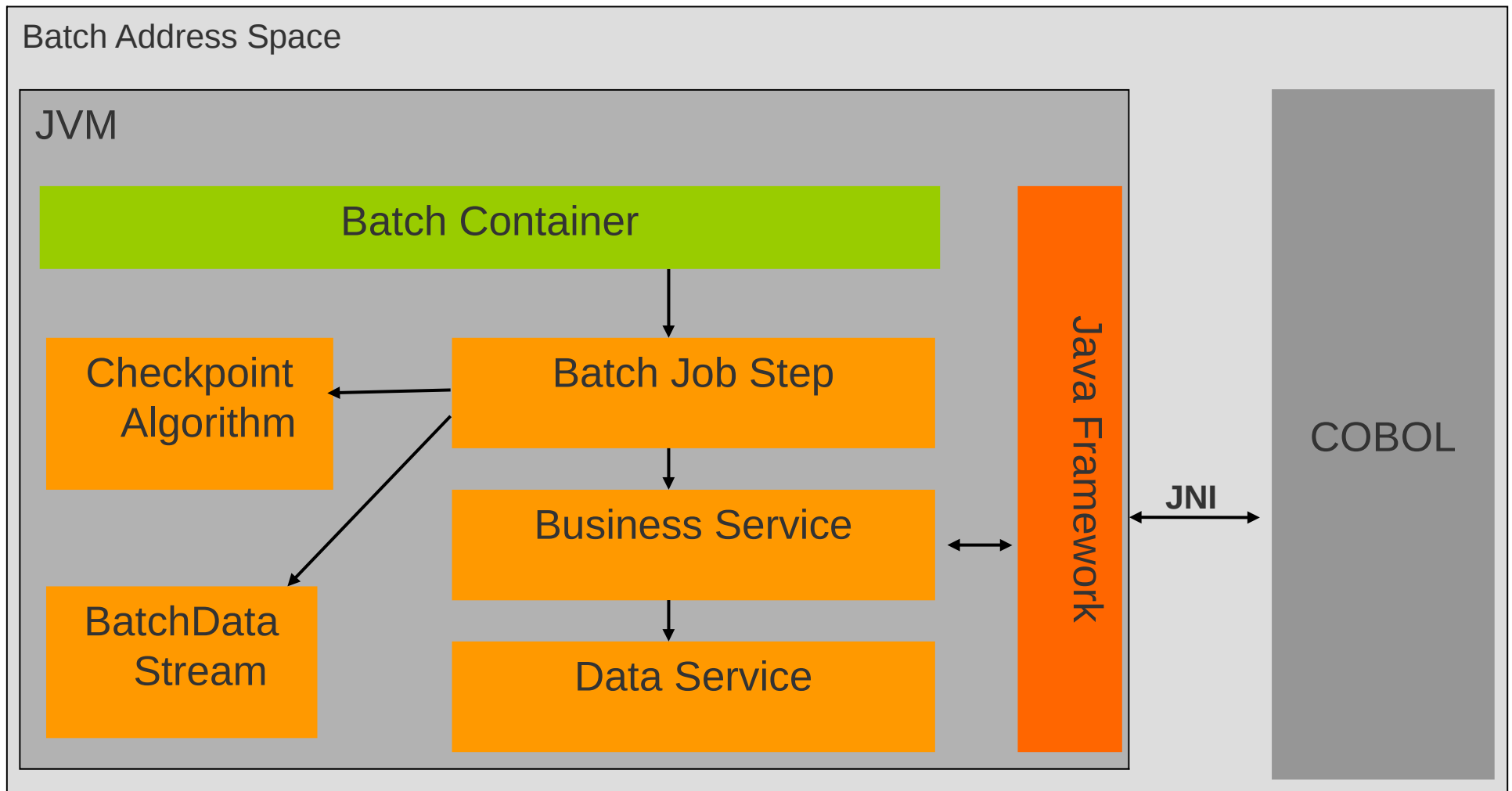
- The framework has to encapsulate the platform specific interfaces:
 - Database Access
 - Connection Management
 - Program callingwhich are provided by:
 - JEE
 - CICS
 - Batch
- The business logic itself resides in plain java objects (POJO)
- The framework needs to define specific a specific interface, that matches the requirements of the applications
- The definition of that framework is sometimes already done by taking the JEE interface and design a testing framework for workstations



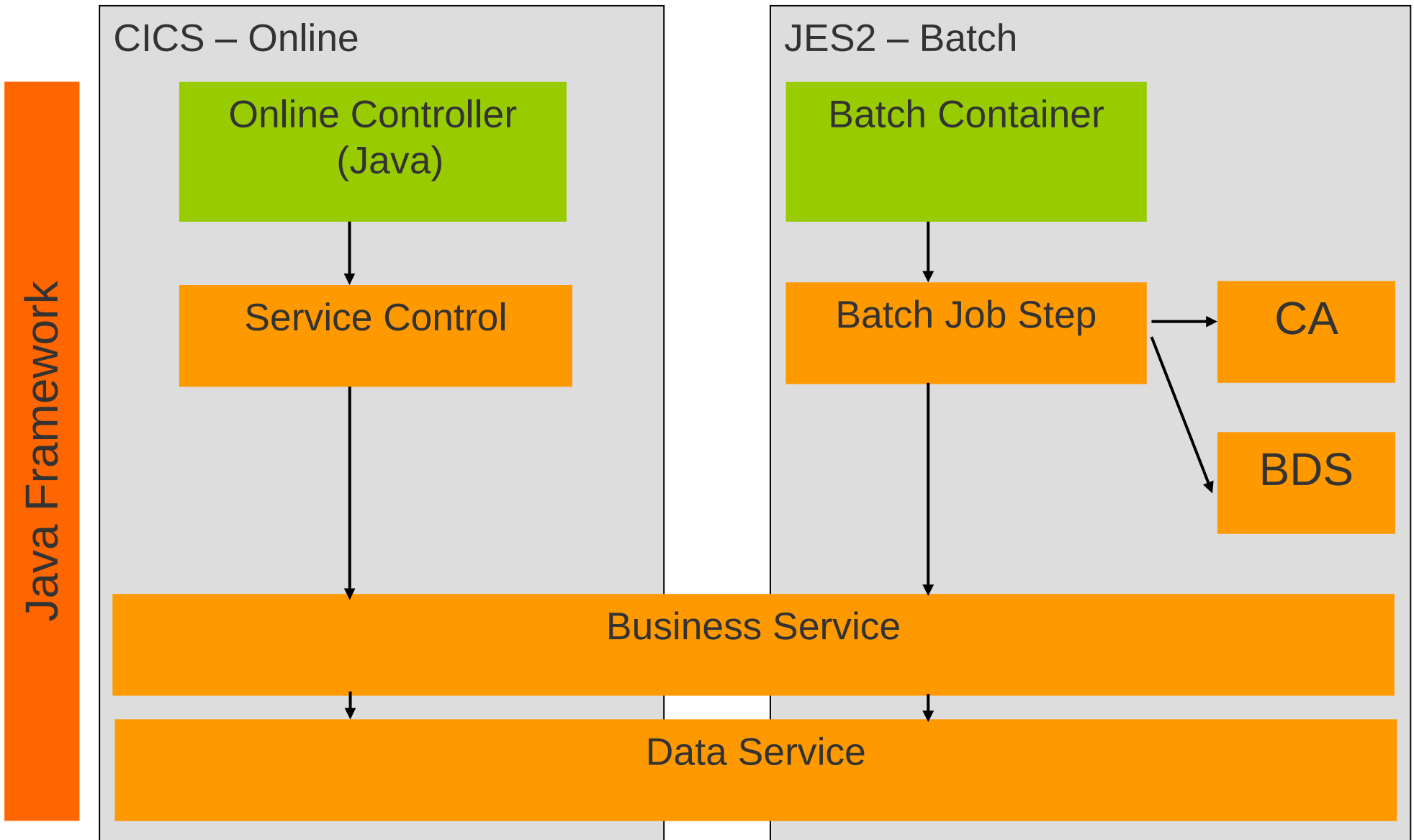
Online Architecture (CICS)



Batch Architecture



Online and Batch Architecture together



A sample scenario may be the extension of a Cobol based application with Java modules

