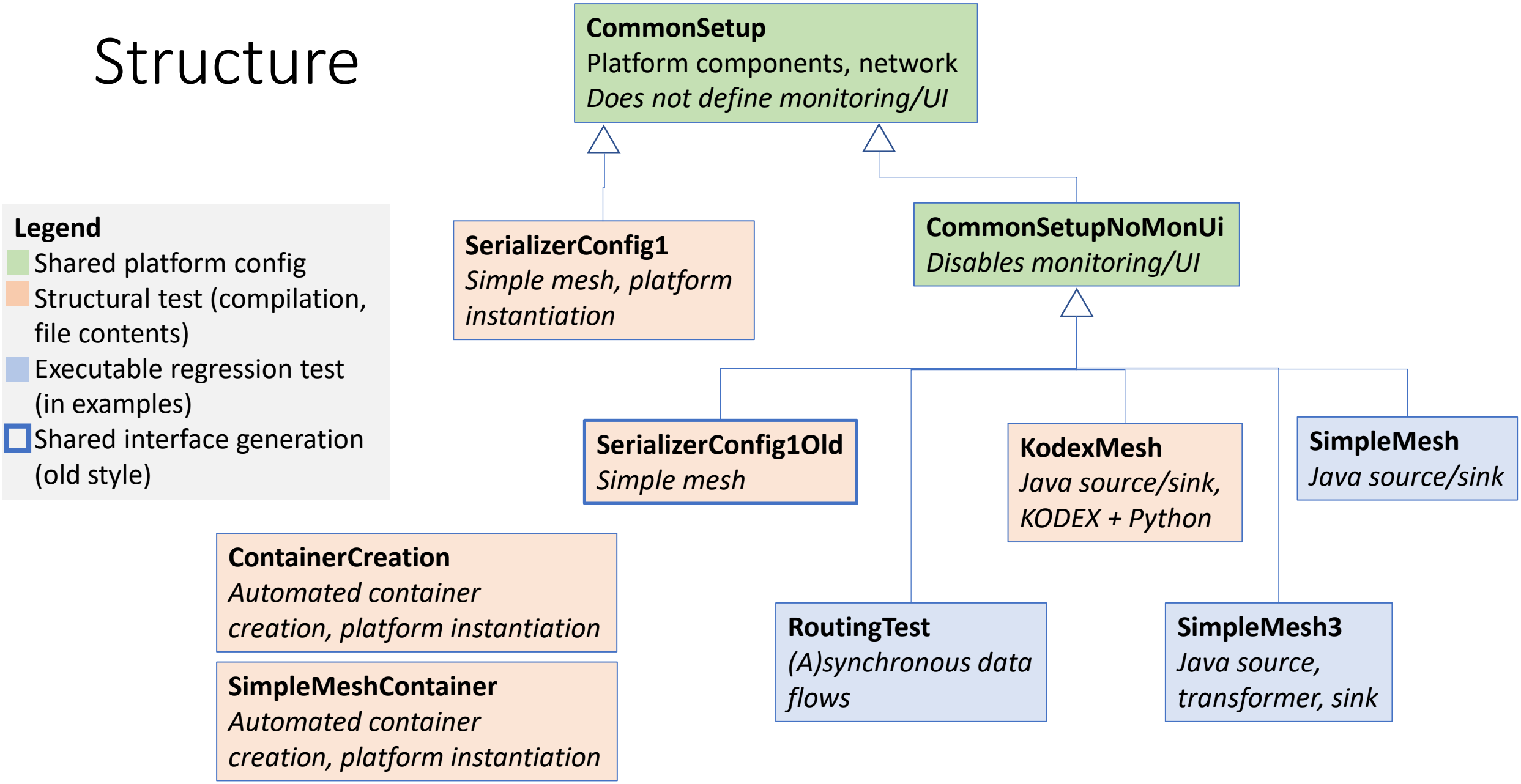
Purpose

- Structural regression tests
  - Does the build process work
  - Is source code compilable (Java, Python)
  - Is Yaml structure ok
  - Are generated files not empty
  - Are expected files there
- Preparation for execution regression tests (in dependent project `examples`)
- Mutual dependency on test.configuration.configuration implementing the services (based on generated interfaces, integrated via dynamic class loading)

Structure

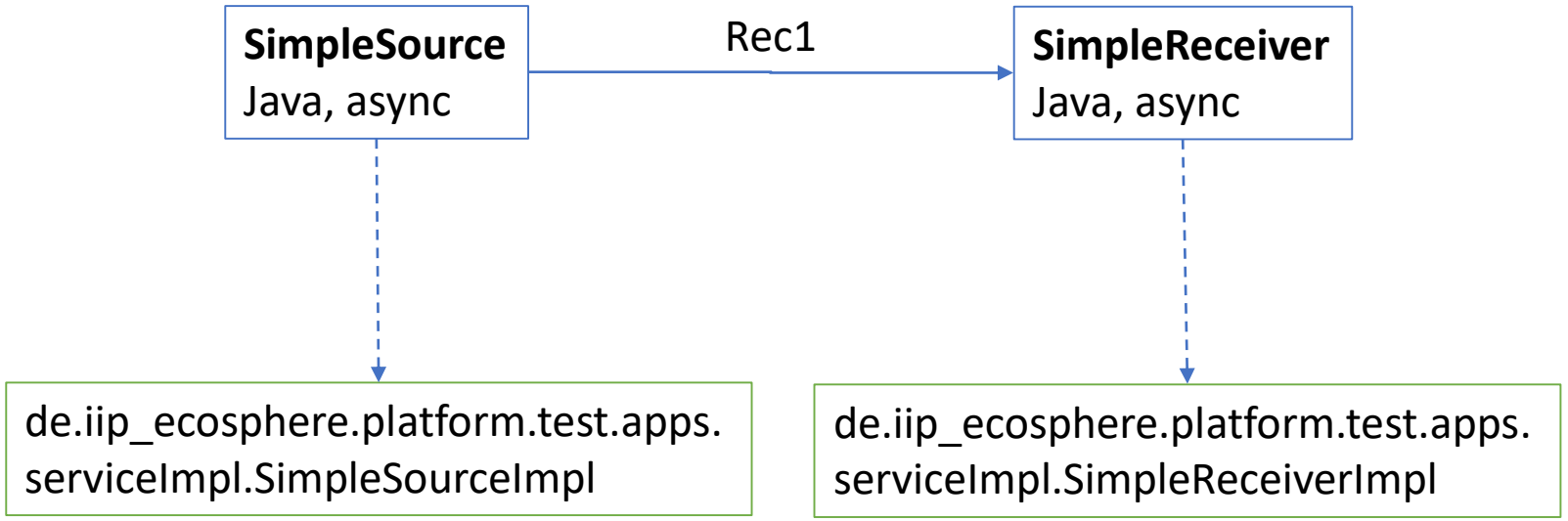

CommonSetup
Platform components, network
Does not define monitoring/UI
SerializerConfig1
Simple mesh, platform instantiation
CommonSetupNoMonUi
Disables monitoring/UI
SerializerConfig1Old
Simple mesh
KodexMesh
Java source/sink, KODEX + Python
SimpleMesh
Java source/sink
RoutingTest
(A)synchronous data flows
SimpleMesh3
Java source, transformer, sink
ContainerCreation
Automated container creation, platform instantiation
SimpleMeshContainer
Automated container creation, platform instantiation
Legend
Shared platform config
Structural test (compilation, file contents)
Executable regression test (in examples)
Shared interface generation (old style)

SimpleMesh

- Purpose:
  - Simple data flow among two services (source, sink)
  - Asynchronous communication
- Used also in the Install Package
- Used as regression test in `examples`

SimpleMesh

In test.configuration.configuration
de.iip-ecosphere.platform:
apps.ServiceImpl:
iipVer

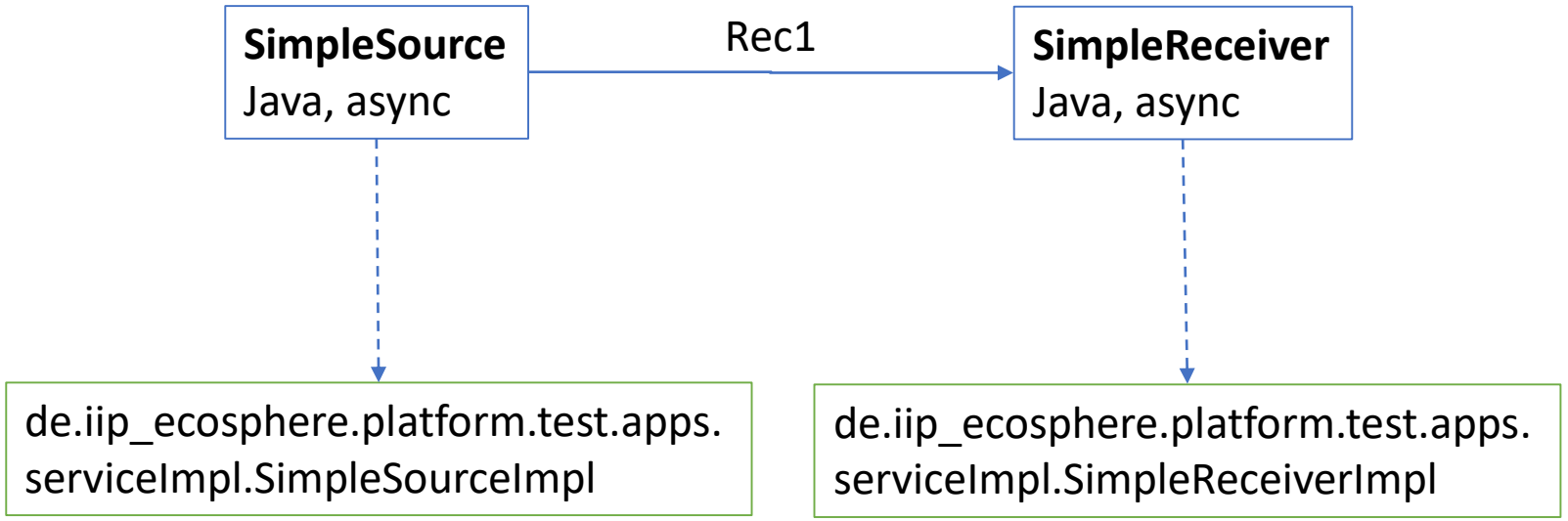


SimpleMeshContainer

- Like SimpleMesh, but with Container creation enabled

SimpleMeshContainer

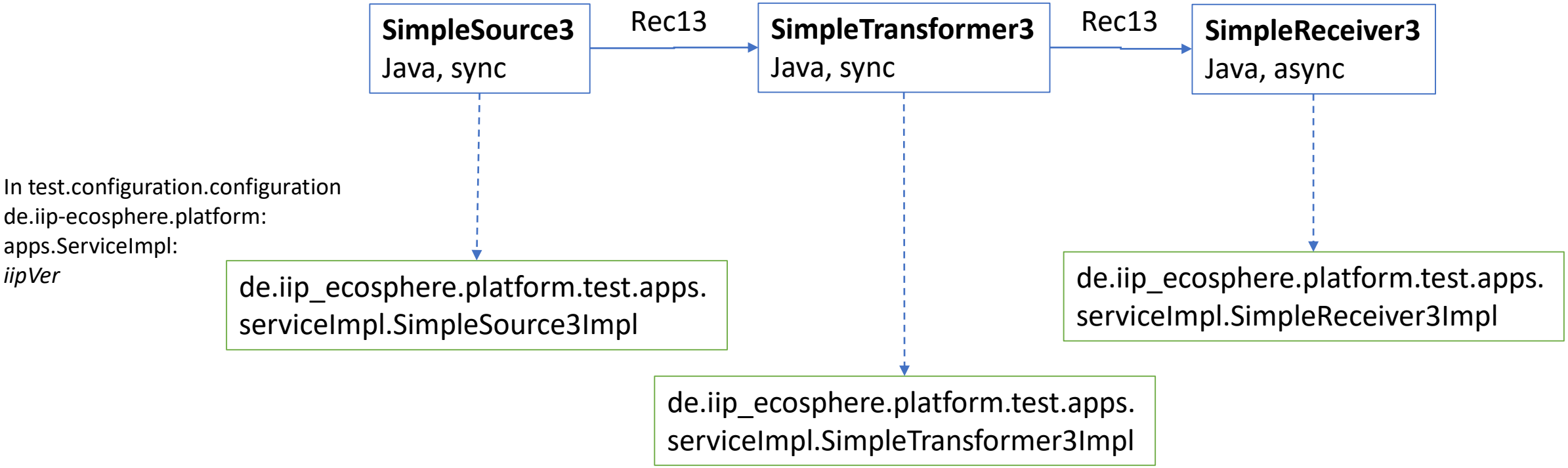
In test.configuration.configuration
de.iip-ecosphere.platform:
apps.ServiceImpl:
iipVer



SimpleMesh3

- Purpose:
  - Simple data flow among three services (source, transformer, sink)
  - Mostly synchronous communication
- Used as regression test in `examples`

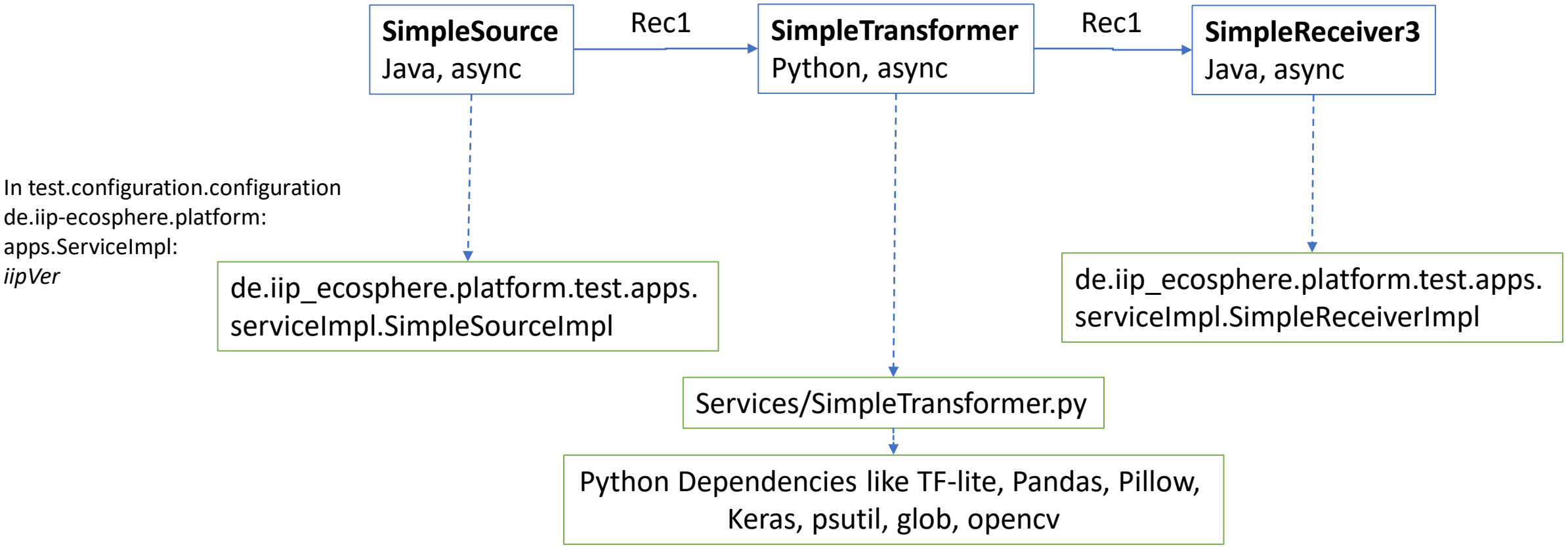
SimpleMesh3 (simple flow with transformer, partially sync)



ContainerCreation

- Based on SimpleMesh, also like SimpleMesh3 but with Python service
- Container creation with Python dependencies

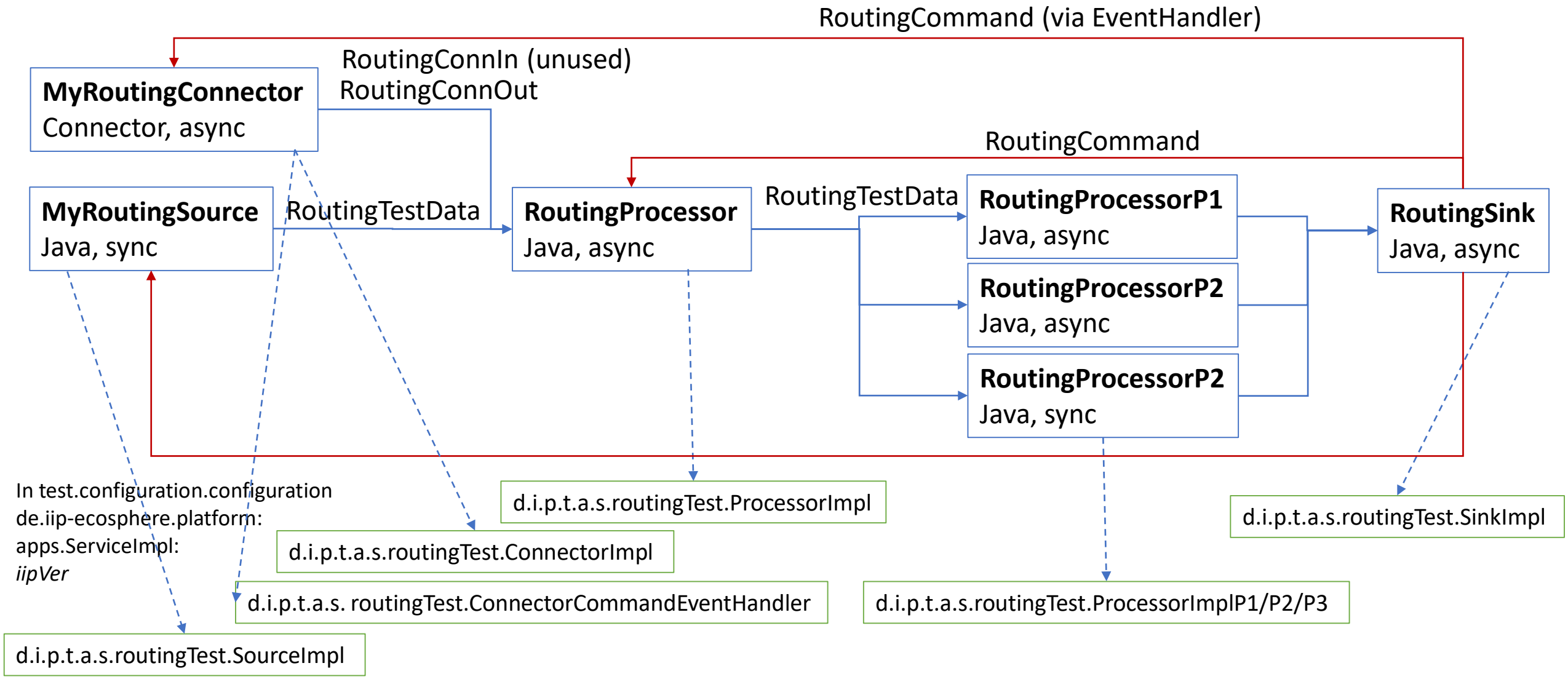
ContainerCreation



RoutingTest

- Purpose:
  - Forward and backward flows
  - Synchronou source
  - Handwritten connector reacting on backward flow
  - Parallel asynchronous paths
- Used as regression test in `examples`

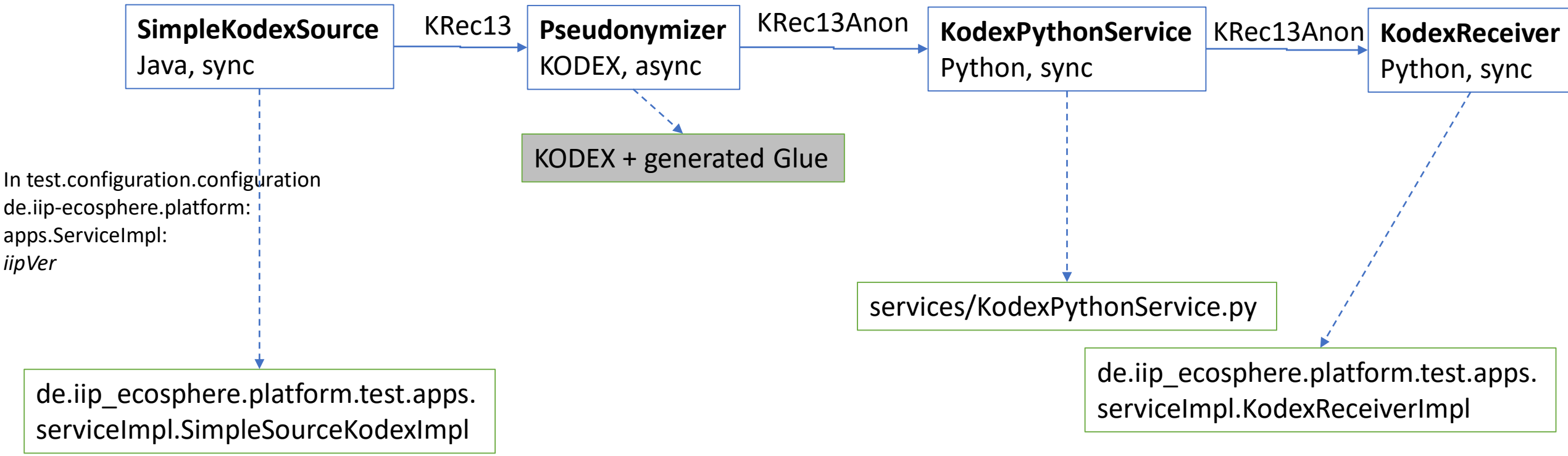
RoutingTest



KodexMesh

- Purpose:
  - Test KODEX integration
  - Later: Also Python, synchronous
- Purely structural test

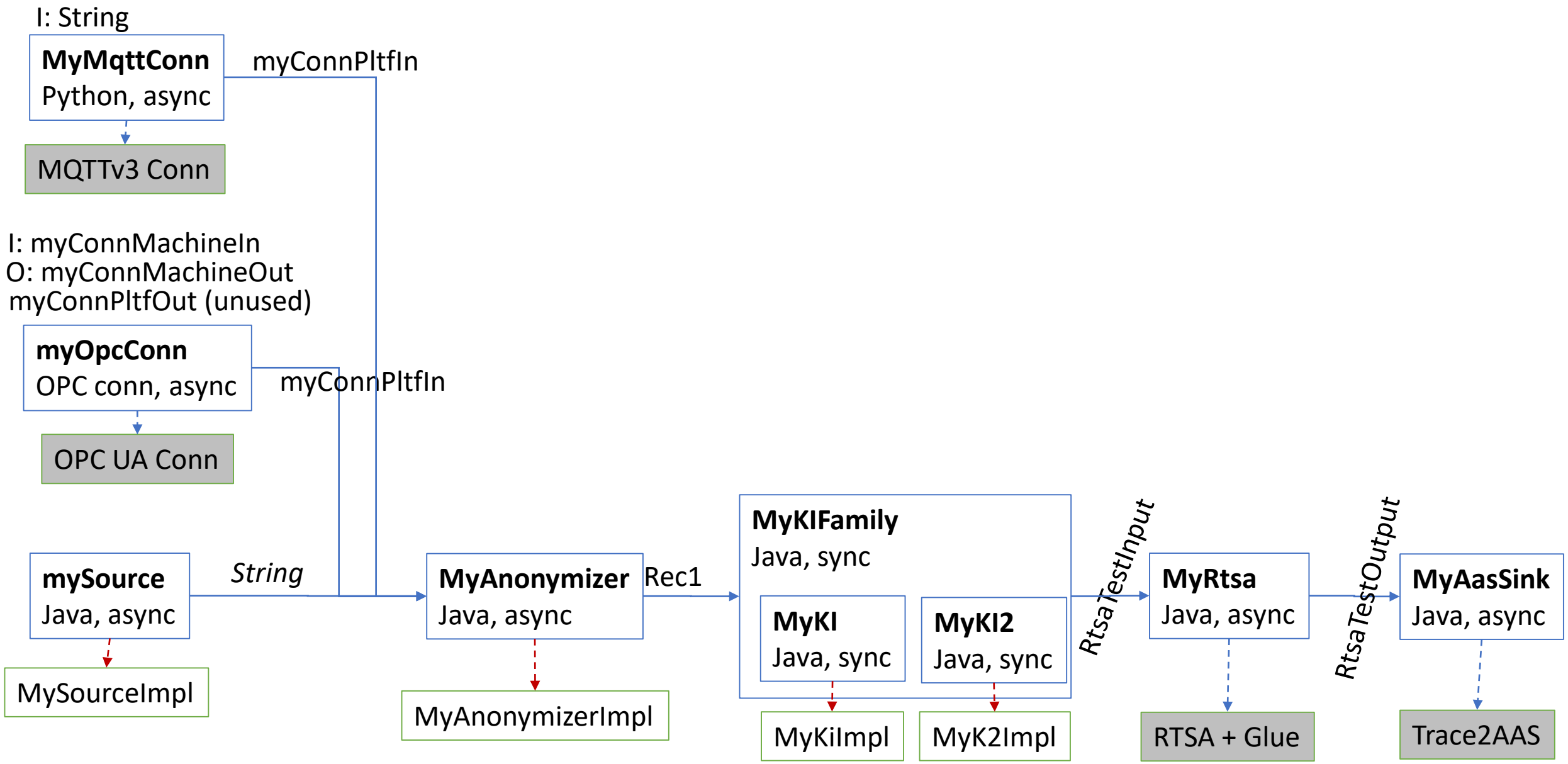
KodexMesh


SimpleKodexSource
Java, sync
KRec13
Pseudonymizer
KODEX, async
KRec13Anon
KodexPythonService
Python, sync
KRec13Anon
KodexReceiver
Python, sync
KODEX + generated Glue
In test.configuration.configuration
de.iip-ecosphere.platform:
apps.ServiceImpl:
iipVer
services/KodexPythonService.py
de.iip_ecosphere.platform.test.apps.
serviceImpl.SimpleSourceKodexImpl
de.iip_ecosphere.platform.test.apps.
serviceImpl.KodexReceiverImpl

SerializerConfig1

- Name: From original test that tested generation of a serializer
- Purpose:
  - Two connectors with data transformation
  - RTSA integration
  - Service family
  - Service parameters
- Purely structural test (not executable, no implementations attached)

SerializerConfig1



SerializerConfig1-old

- Name:
  - From original test that tested generation of a serializer
  - Derived from SerializerConfig1
  - Old-style not-separated interfaces
- Purpose:
  - Two connectors with data transformation
  - Simple data chain
- Purely structural test (not executable, no implementations attached)

SerializerConfig1-old (connector gen, impl. irrelevant)