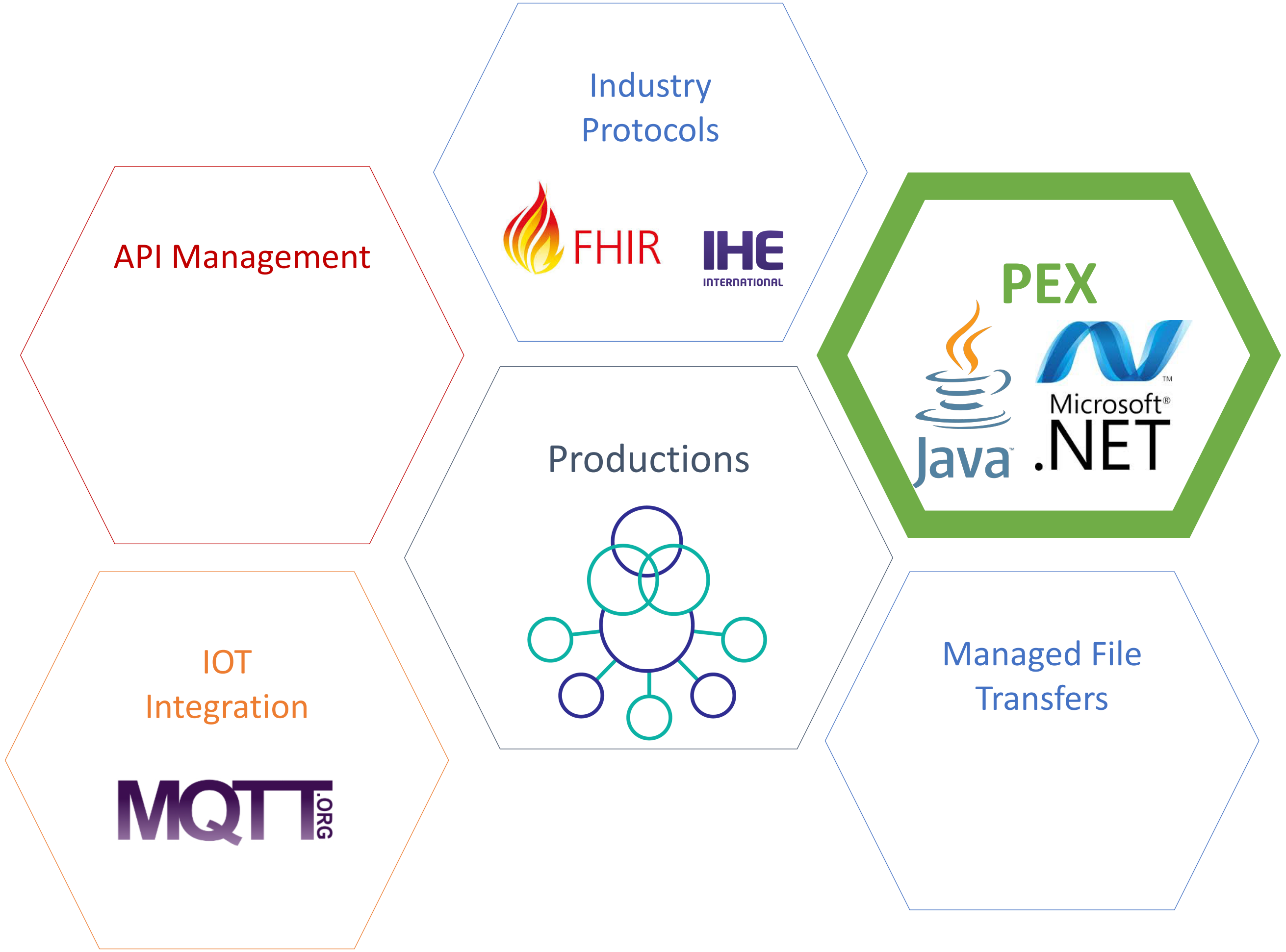
PEX (Production Extension Framework)

Workshop

Alberto Fuentes
Sales Engineer



Industry Protocols

FHIR IHE INTERNATIONAL

API Management

PEX

Java Microsoft® .NET

Productions

IOT Integration

MQTT.ORG

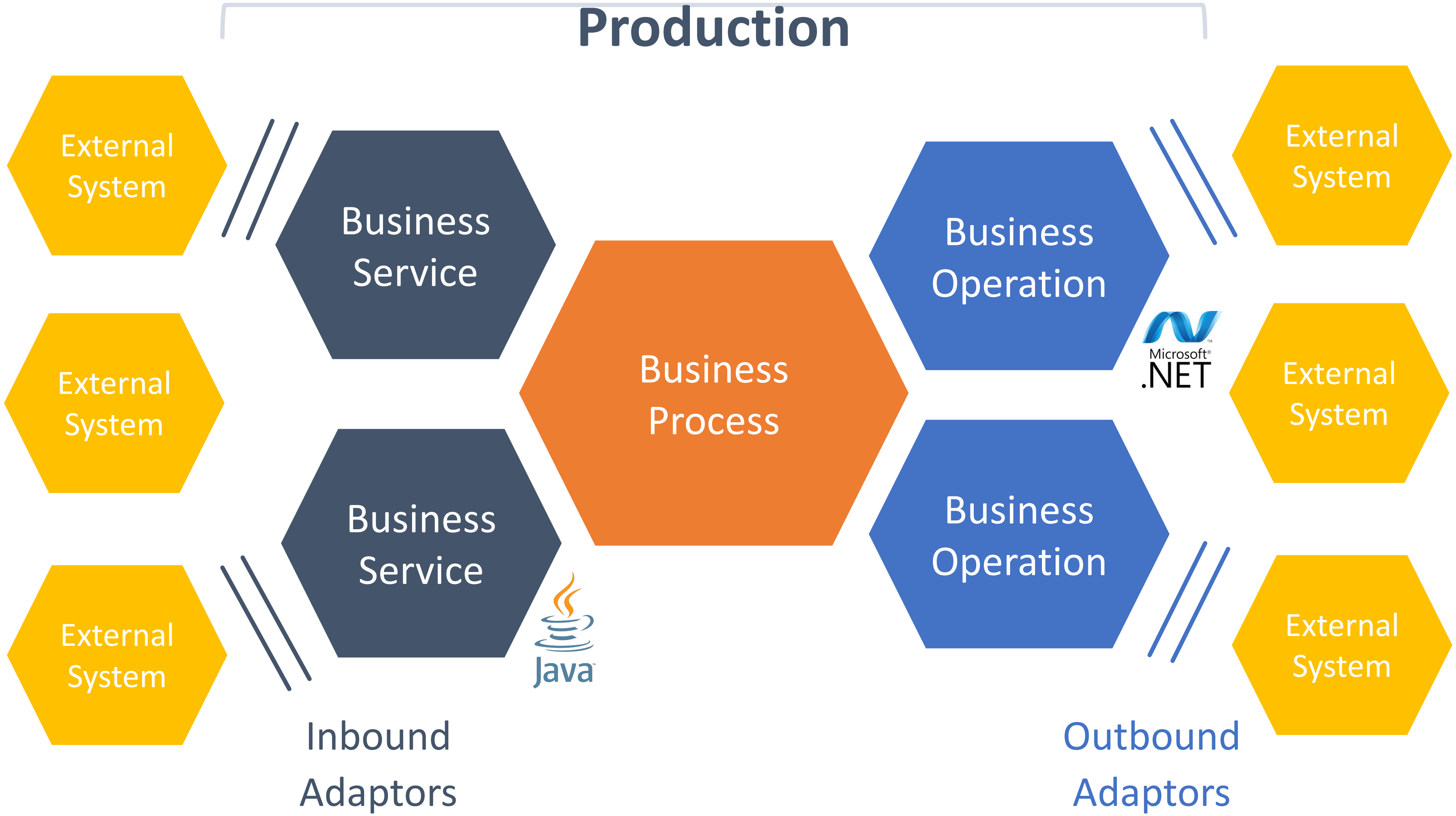
Managed File Transfers

What is PEX ?

PEX: Production Extension Framework

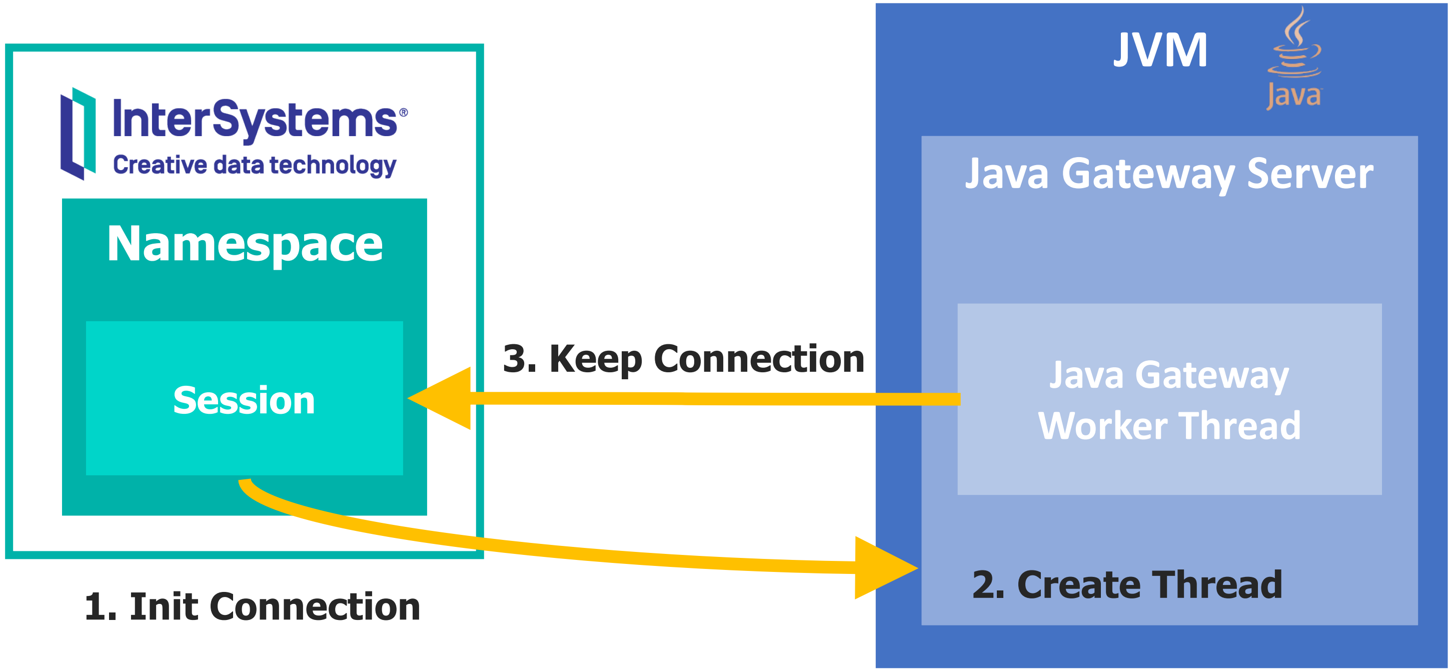
- Choice of developing language in interoperability productions:
  - Supported: Java, .NET.
- You can implement any production component:
  - Business Services, Business Processes, Business Operations.
  - Messages.
  - Inbound / Outbound Adapters.
- Allows you to:
  - Create an **entire production** in Java or .NET.
  - Create a production that has a **mix** of Java, .NET, ObjectScript components.
  - Creating new adapters for protocols using available **Java or .NET libraries**.
  - Combine Java, .NET components with a **robust interoperability framework** (persistent message tracing, alert handling, industry protocols like HL7, ASTM, FHIR, DICOM, etc.)

What is PEX ?


Production
External System
External System
External System
Business Service
Business Service
Java
Business Process
Business Operation
Business Operation
Microsoft .NET
External System
External System
External System
Inbound Adaptors
Outbound Adaptors

Java / .NET Gateway

- Gateway servers executed on Java / .NET (native components included in IRIS).
- IRIS connects to Gateway using TCP/IP. Gateway
- Gateway executes Java / .NET code (bi-directional).
- Gateway server can run on a remote host.
- Gateway can be started from portal, production, container.



How it works ?

Basic Workflow

1 **Develop Java / .NET component**

· Use provided IRIS libraries.
· Generate executable code.

2 **Add PEX components in IRIS production**

· PEX components are pre-built.
· Configure components to connect to Gateway and use your code.

3 **Start Gateway Server & Run**

· Start your Gateway from Portal, Production or a container.
· Run your interoperability productions using your components.

Libraries

- Libraries to include in your project.

<irisdir>/dev/java/lib/JDK18/
intersystems-gateway-3.1.0.jar
intersystems-jdbc-3.1.0.jar
gson-2.8.5.jar

Implementation

- **`com.intersystems.enslib.pex.`**
- BusinessService, BusinessProcess, BusinessOperation
- InboundAdapter, OutboundAdapter, Message.

PEX
(Java)

Libraries

- Libraries to include in your project.

<irisdir>\dev\dotnet\bin\v4.5\
`InterSystems.Data.Gateway64.exe`
`InterSystems.Data.IRISClient.dll`

Implementation

- **`InterSystems.EnsLib.PEX.`**
- BusinessService, BusinessProcess, BusinessOperation
- InboundAdapter, OutboundAdapter, Message.

PEX (.NET)

PEX Component Life cycle

Initialization

- `OnInit()`

Running

- **Business Service**: `OnProcessInput()`
- **Business Process**: `OnRequest(), OnResponse(), OnComplete()`
- **Business Operation**: `OnMessage()`

Shutdown

- `OnTearDown()`

Sending Messages

- `SendRequestSync()`
- `SendRequestAsync()`

Log into IRIS Event Log

- `LOGINFO()`
- `LOGALERT()`
- `LOGWARNING()`
- `LOGERROR()`

PEX Utils

PEX Component Settings

Pre-Built components

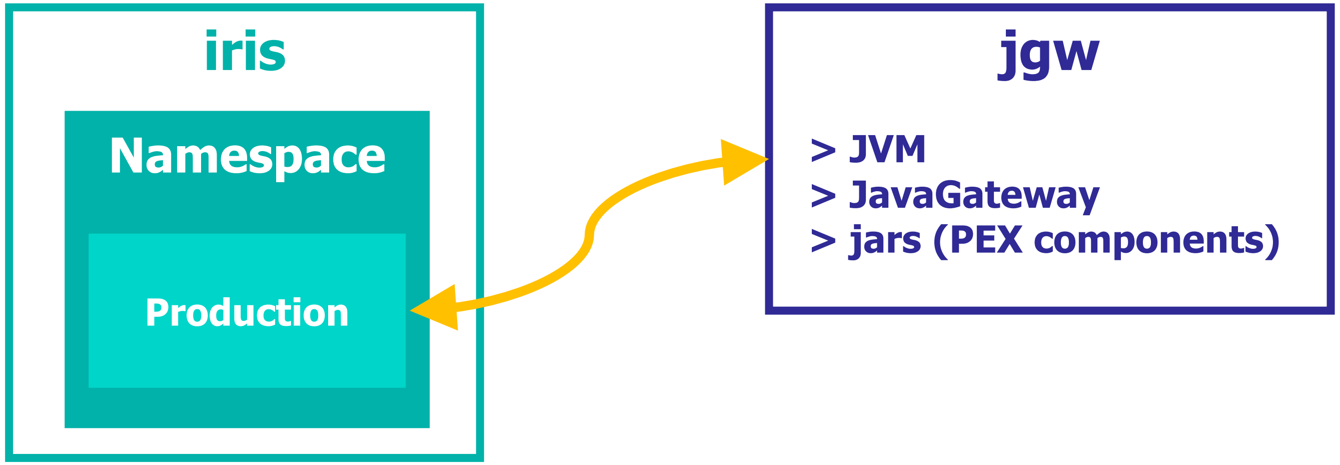
- PEX components for Interoperability productions are already built.
- You only need to configure some settings.

Settings

SETTING	DESCRIPTION
Gateway Host / Port	Address / Port of Java or .NET Gateway
Extra ClassPath	Additional libraries needed for execution
Remote ClassName	The Java or .NET classname of your component
Remote Settings	Your Java or .NET parameter values, using: <Param>=<Value>

Hands-on: FirstDemo

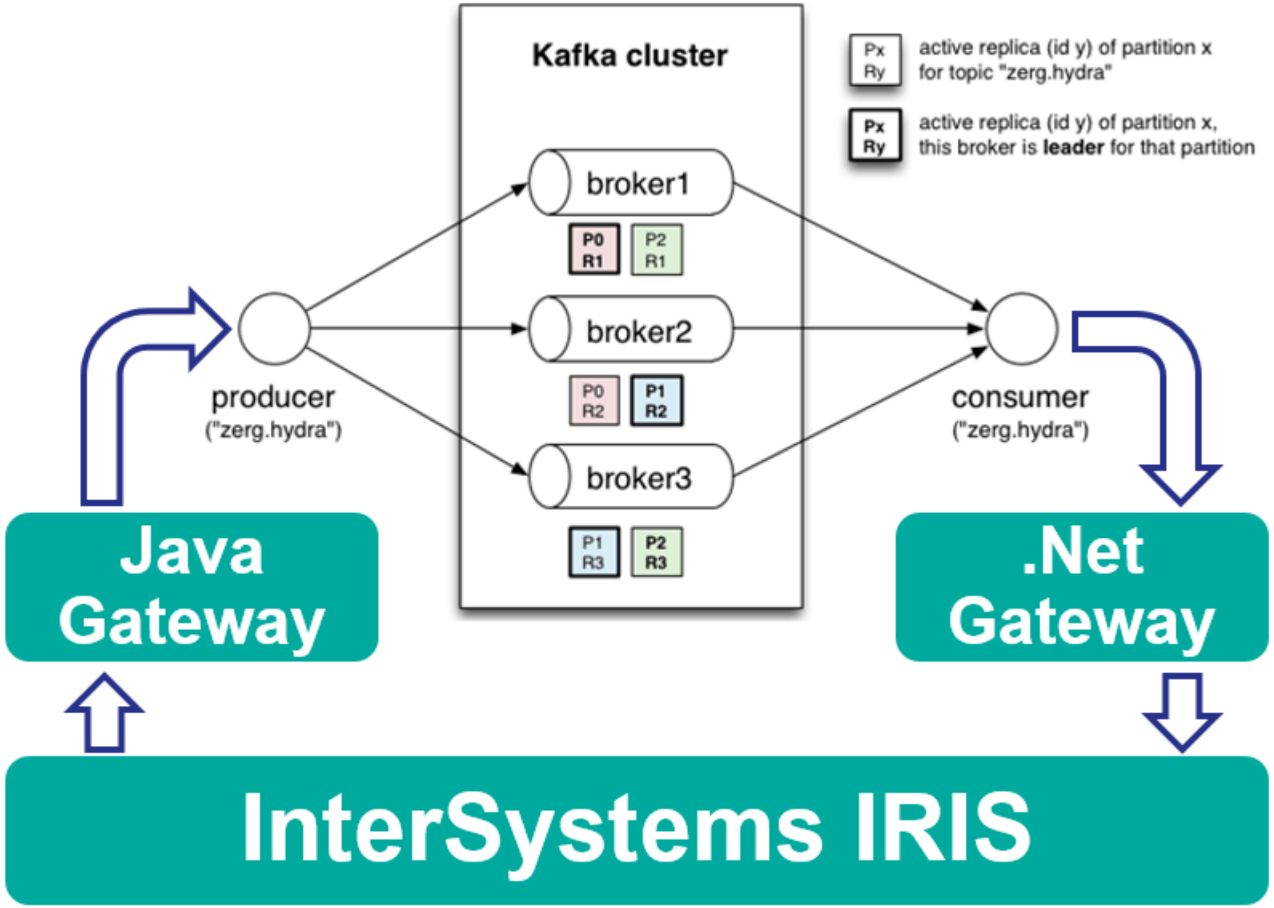
https://github.com/intersystems-ib/workshop-pex



> JVM
> JavaGateway
> jars (PEX components)

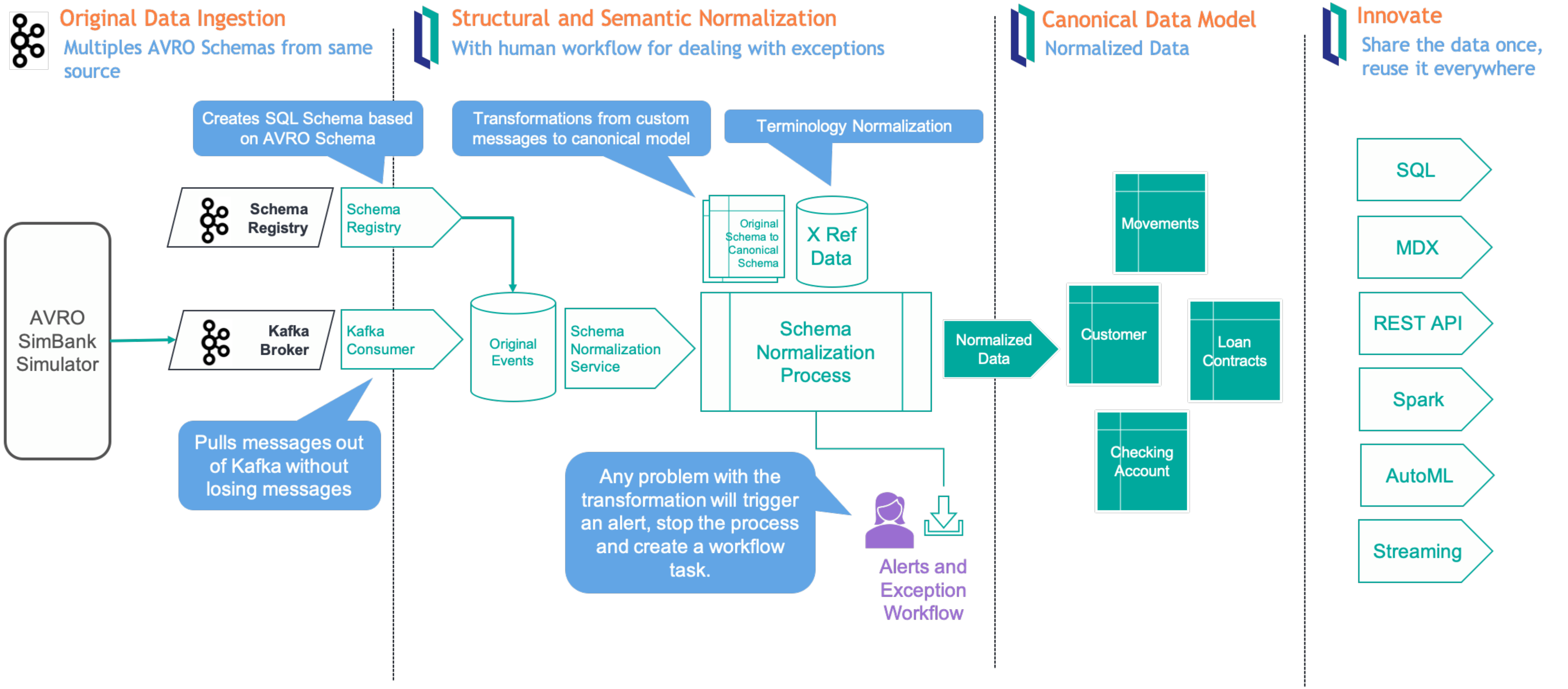
Hands-on

https://github.com/intersystems-community/pex-demo



Explore more complex scenarios

https://github.com/intersystems-community/irisdemo-demo-kafka



Original Data Ingestion

Multiples AVRO Schemas from same source

Structural and Semantic Normalization

With human workflow for dealing with exceptions

Canonical Data Model

Normalized Data

Innovate

Share the data once, reuse it everywhere

Creates SQL Schema based on AVRO Schema

Transformations from custom messages to canonical model

Terminology Normalization

AVRO SimBank Simulator

Schema Registry

Schema Registry

Kafka Broker

Kafka Consumer

Pulls messages out of Kafka without losing messages

Original Events

Schema Normalization Service

Original Schema to Canonical Schema

X Ref Data

Schema Normalization Process

Any problem with the transformation will trigger an alert, stop the process and create a workflow task.

Alerts and Exception Workflow

Normalized Data

Movements

Customer

Loan Contracts

Checking Account

SQL

MDX

REST API

Spark

AutoML

Streaming