

 Seamless

 Integrated

 Consistent

 Personalised

enactor[®]
retail systems for a digital world

Enactor Training Course Toolkit Overview

Toolkit Overview

- Parameter based systems....
- Become complex and virtually untestable
- Adding features and functions time consuming and costly
- New features always require code changes
- The entire solution is understood only by a few
- Code base becomes large and cumbersome
- Process and flow is hard coded

- Build and deploy in days not weeks
- Flexible connectivity
- Design business process first
- Drag functionality and processes like a CAD drawing
- Support distributed development
- Benefits of own build and package
- Stop code being a “black box”
- The “drawing” is the “application”

Applications

Store, Mobile, Clientelling, Stock, Estate Management....

Libraries

Libraries application components for building and enhancing applications ~ 6000 processes built to date

Toolset

Toolset for allowing rapid application development and change

Platform

Built on Digital Technologies, Web Calls, XML, Java

Enactor Enterprise Platform

UI Libraries



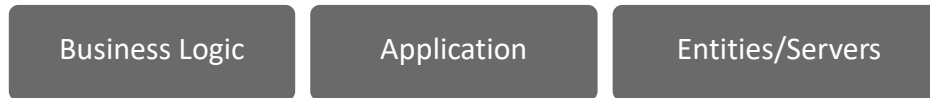
Enactor UI specific jars

Process Flow

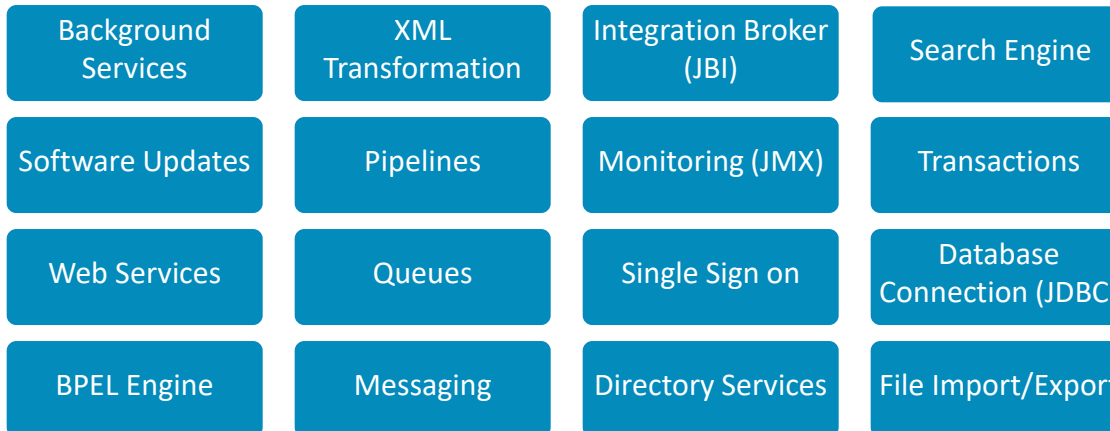


Enactor core UI jars

Model



Services



Enactor core jars

Third Party jars

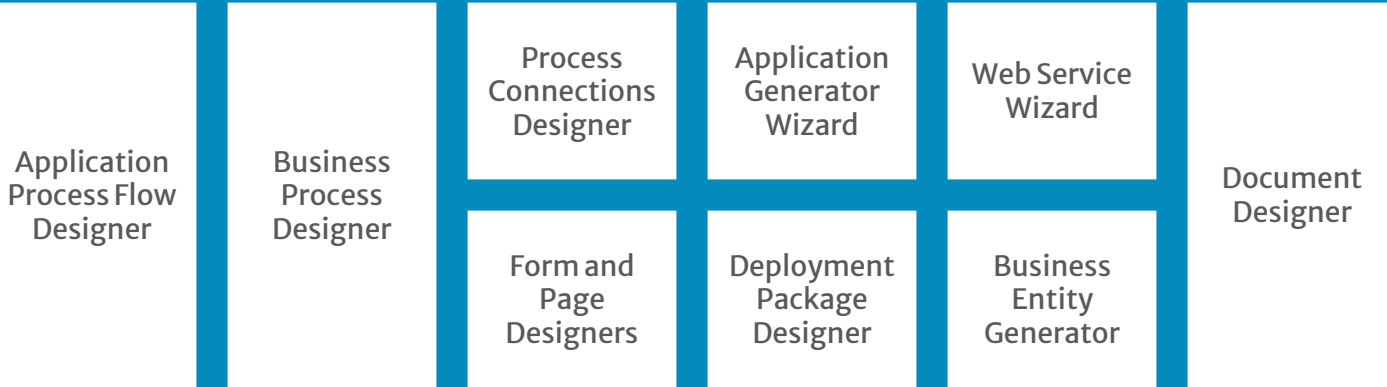
Runtime



Java jars

The Enactor Development Tools

Eclipse

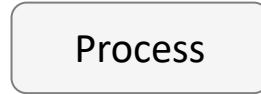


Resource Library

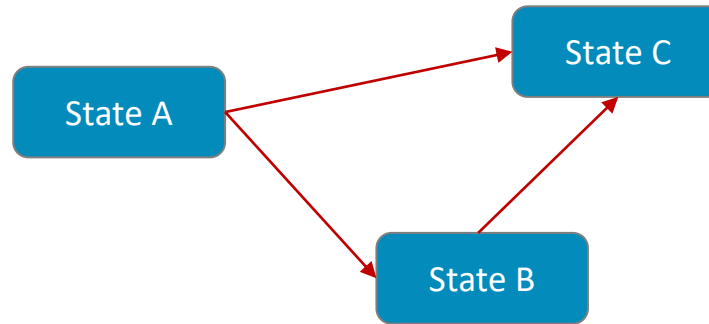
Java Enterprise Platform

- We have found that there are three logical levels of “process”
 - Application Processes control flow and behaviour inside applications
 - BPEL is the technical layer orchestrating the use of Web Services and long running applications
 - Business Processes control flow and interactions between applications and people
- We have developed tools for each of these different levels of process
- Very importantly each tool and runtime is optimised for the level at which they operate

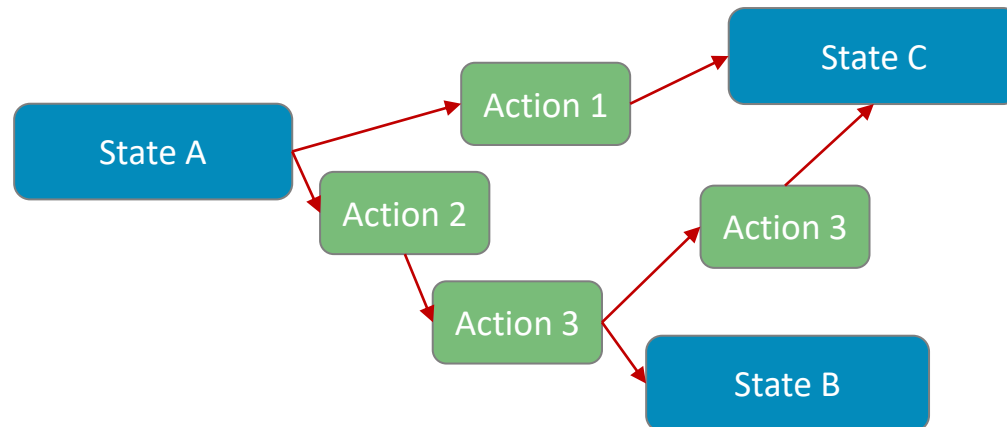
Step 1 – Identify a process



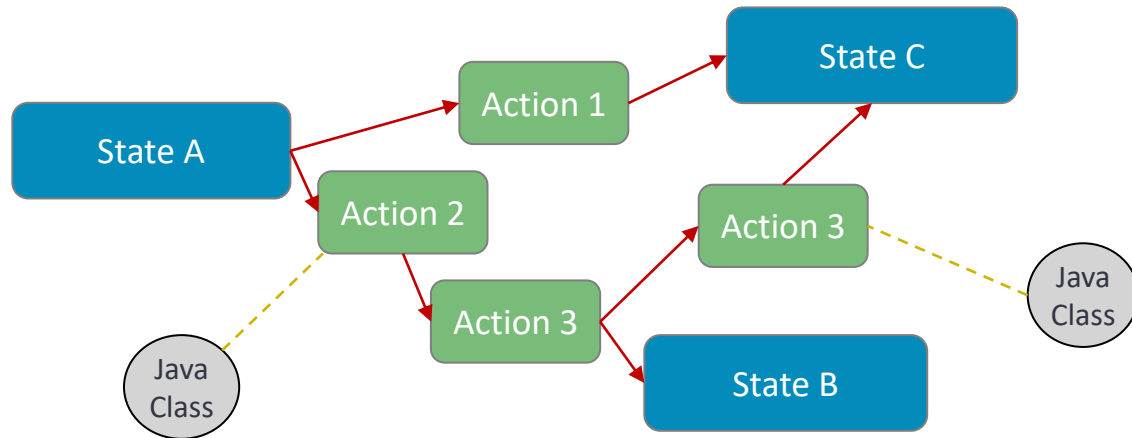
Step 2 – Identify the stages or states in the process and paths



Step 3 – Identify the actions required between states

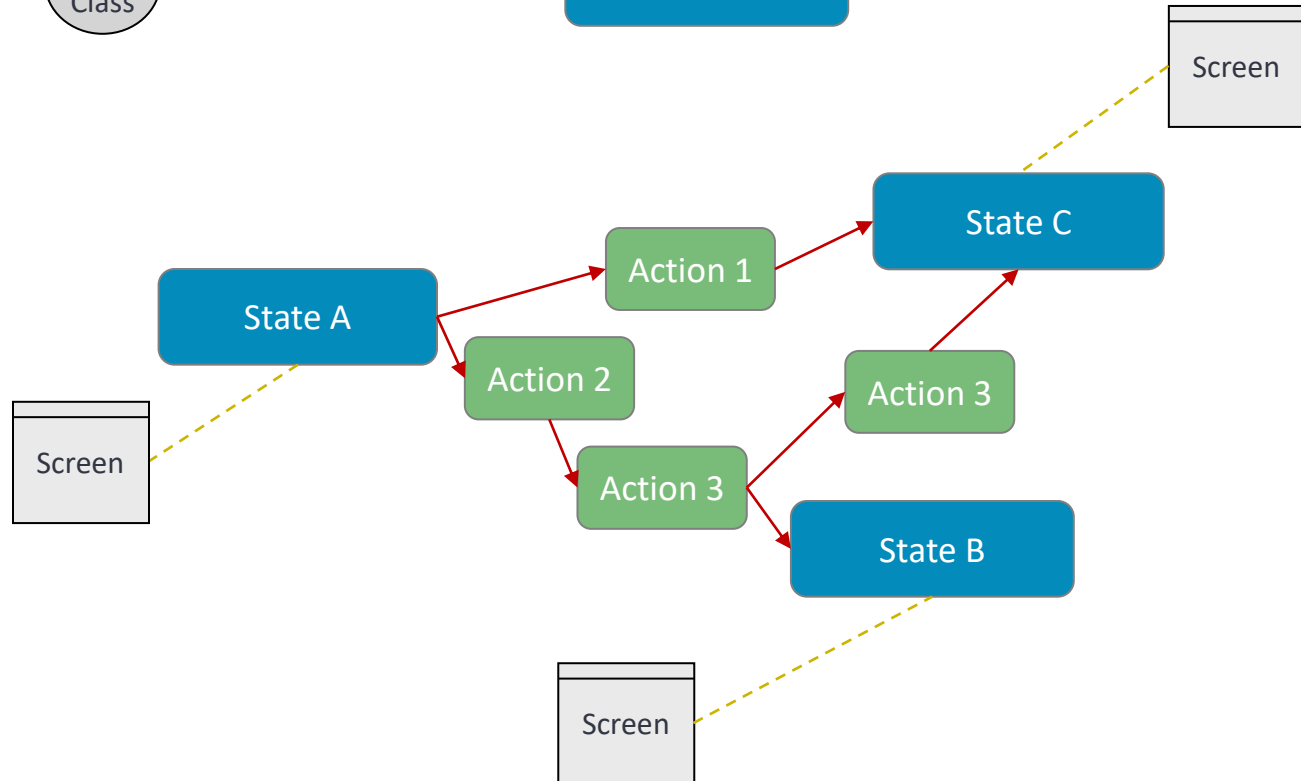


Step 4 – Link the actions to pre-written components



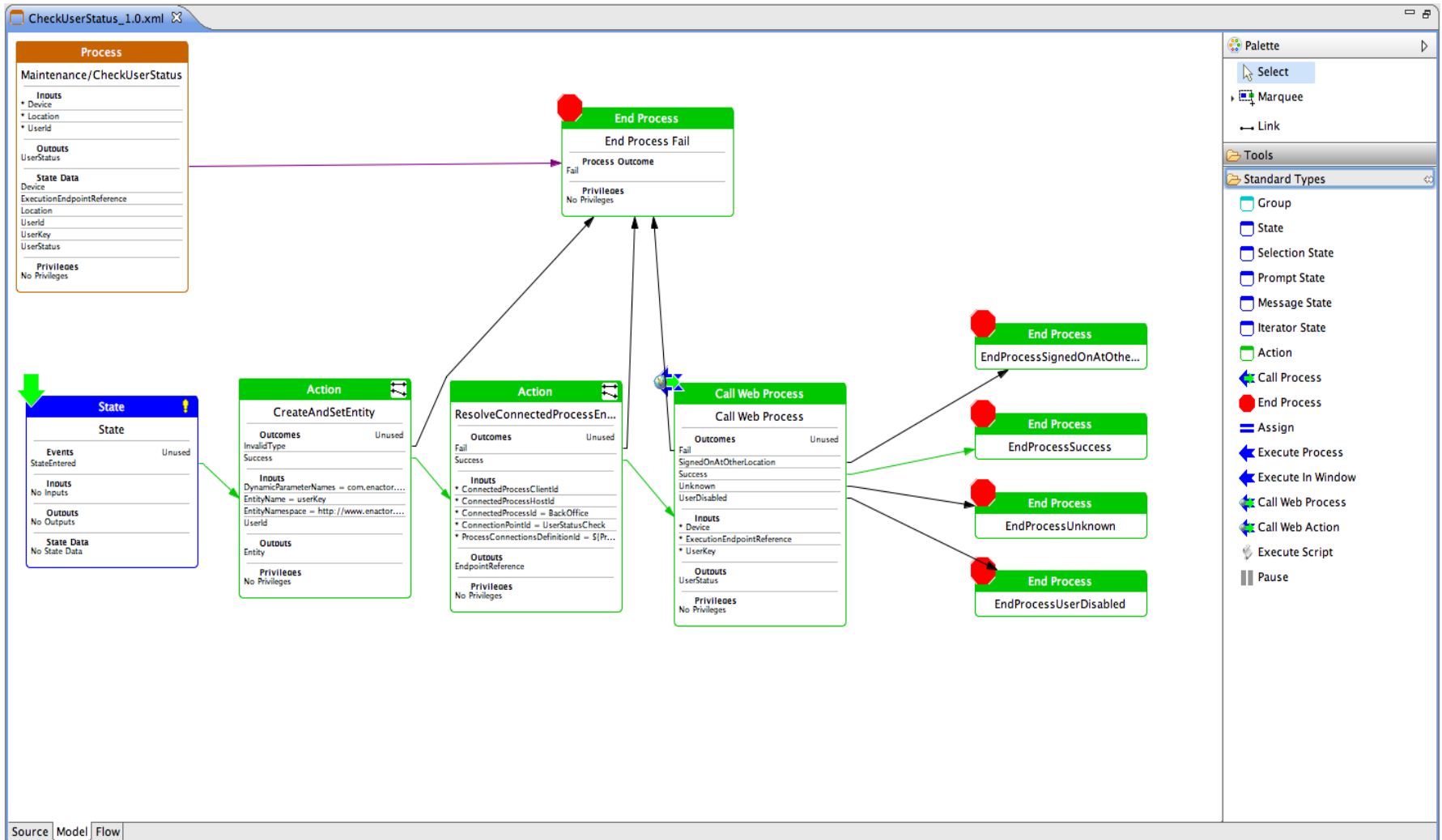
Step 5 – Write new functional components where library not available

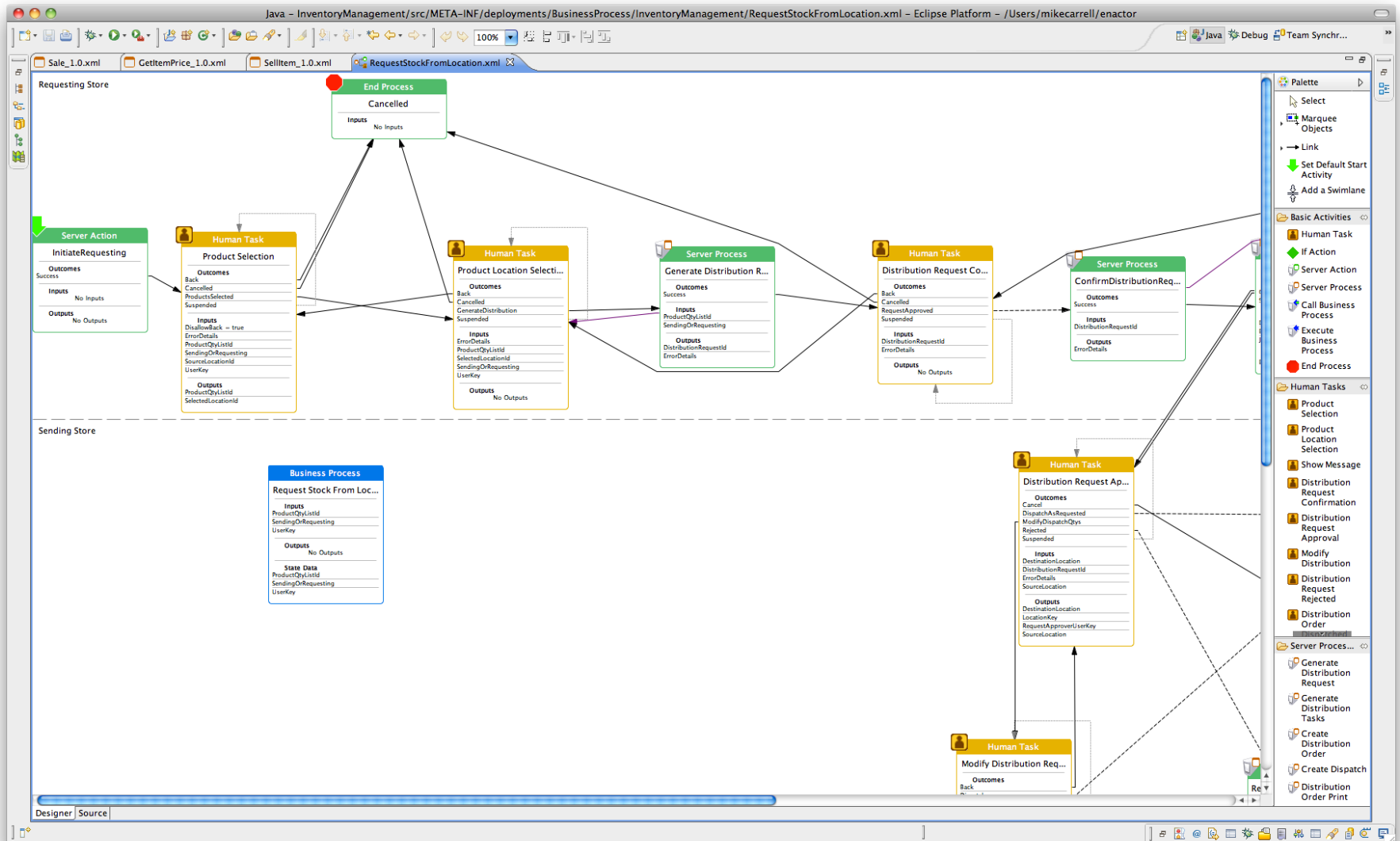
Step 6 – Develop UI screens



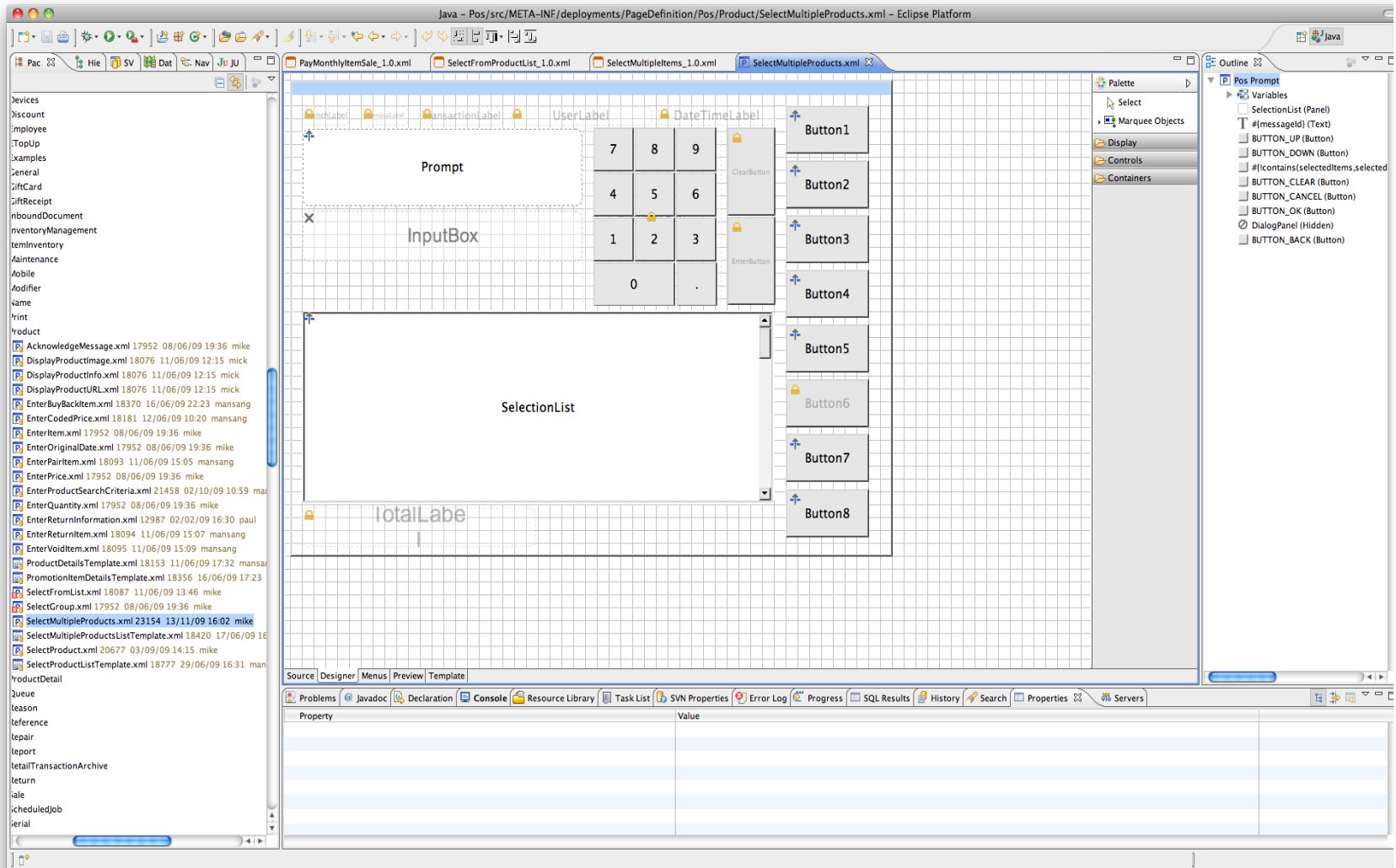
Step 7 – Re-organise states and actions to refine application

Enactor Application Process Designer

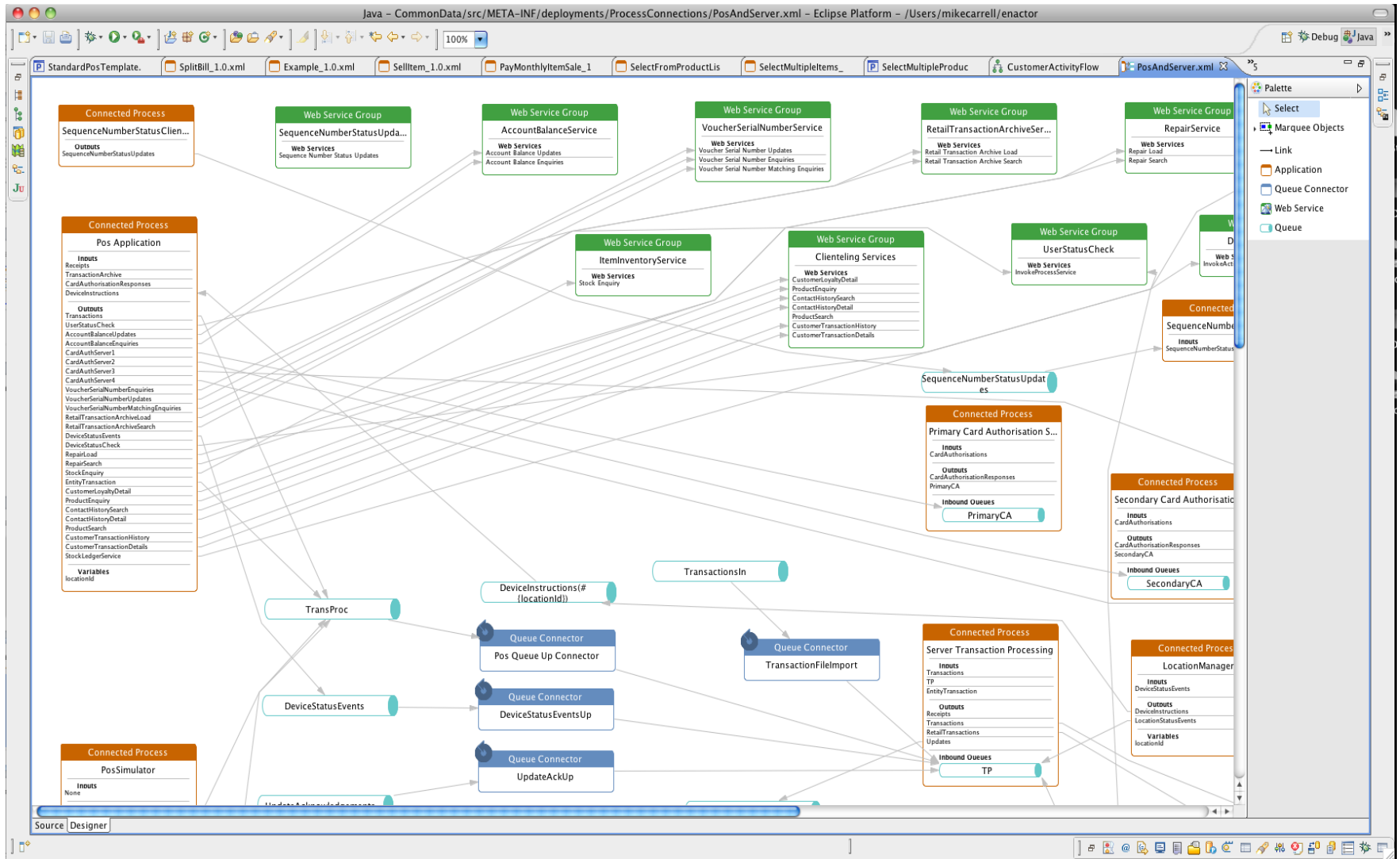


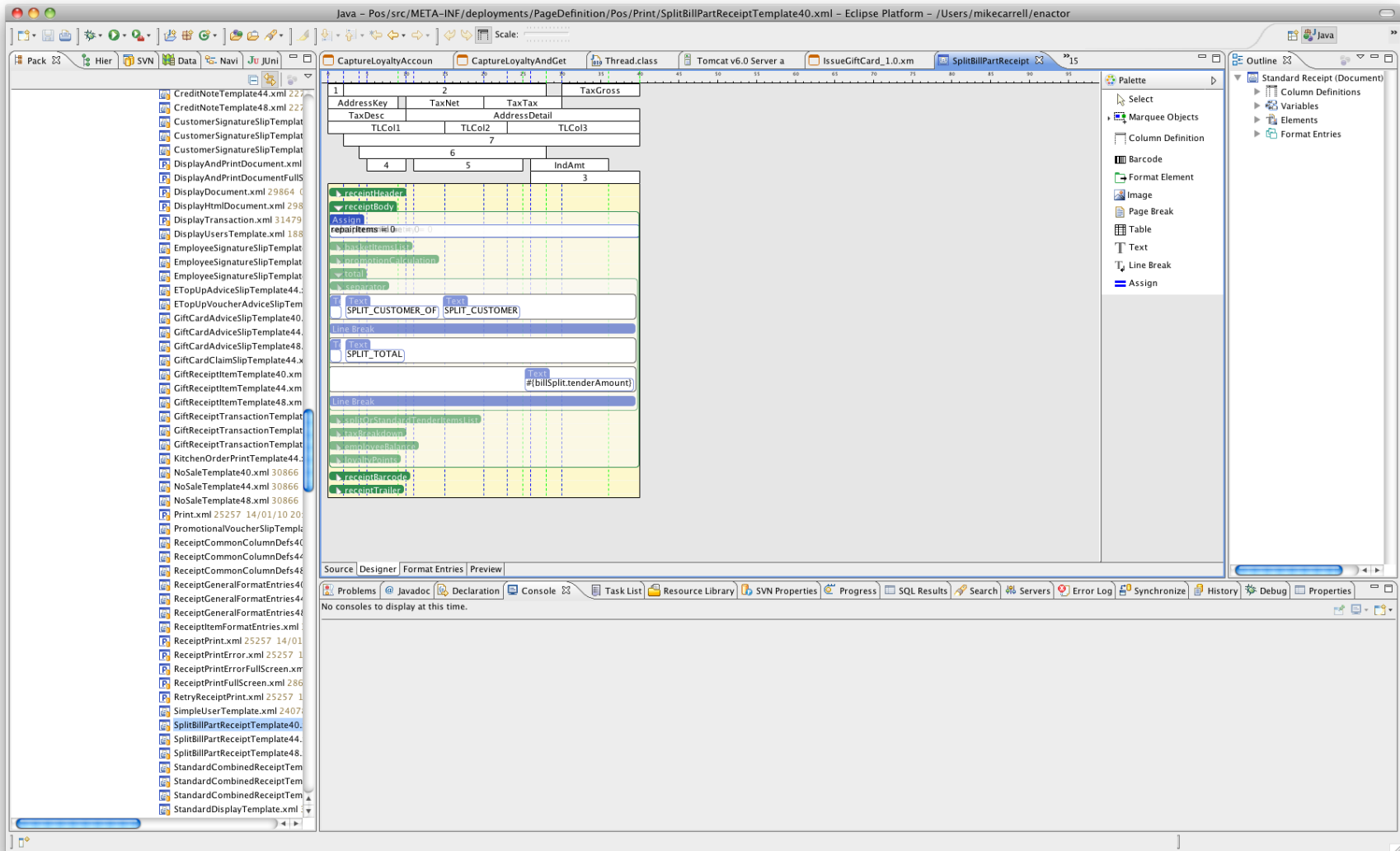


Enactor Page & Form Designer

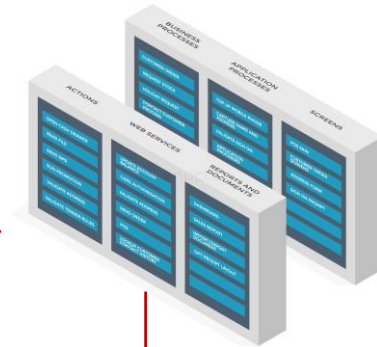


Enactor Connected Processes Designer



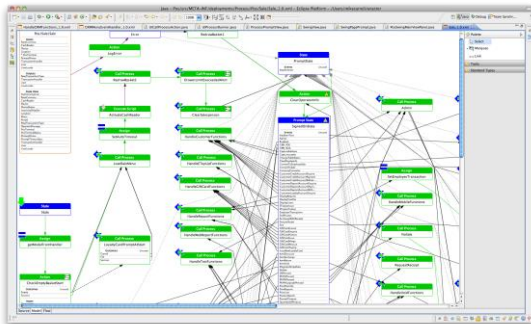


Same code support multiple deployments

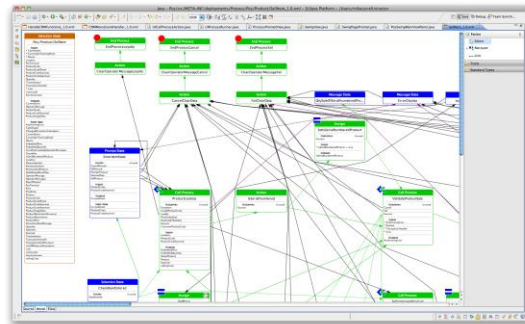


Common Libraries and Services

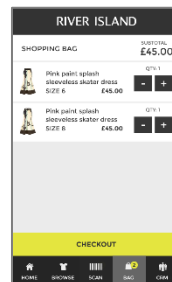
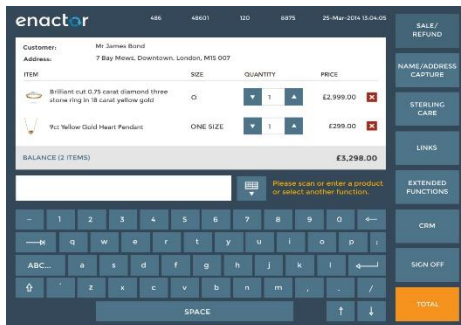
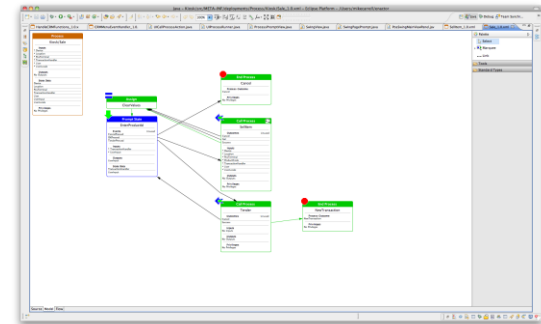
Traditional POS Processes



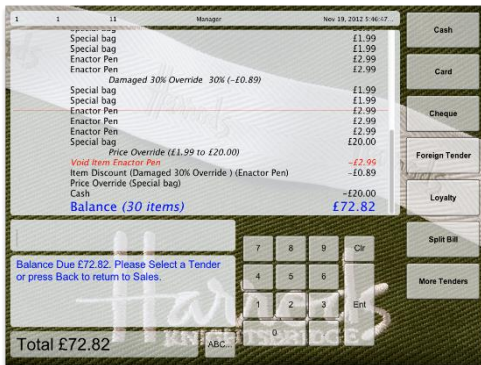
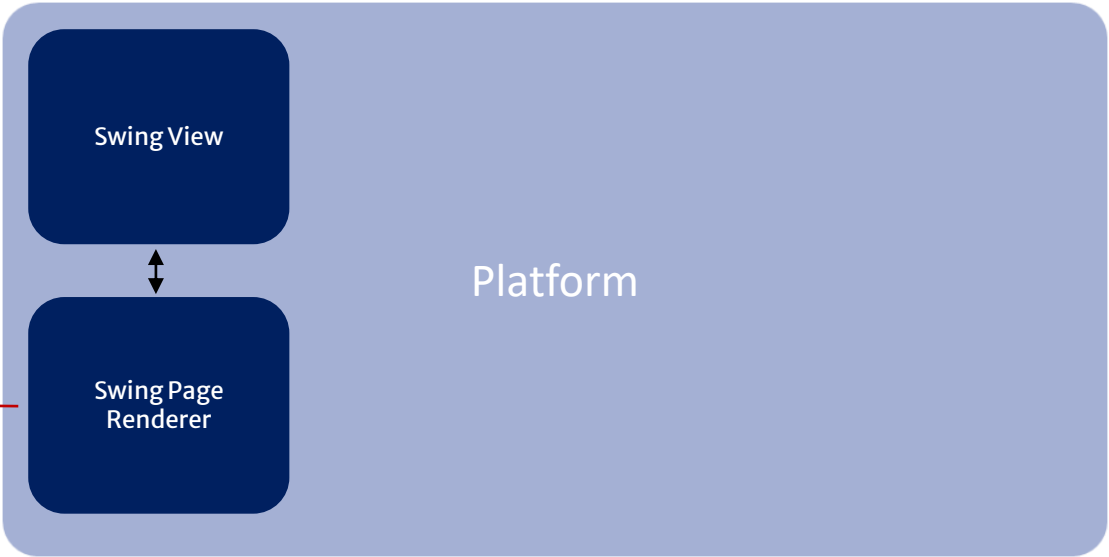
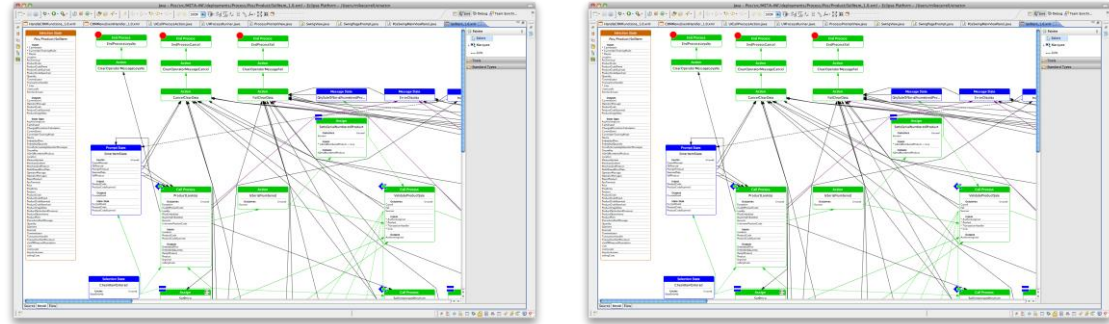
Mobile POS Processes



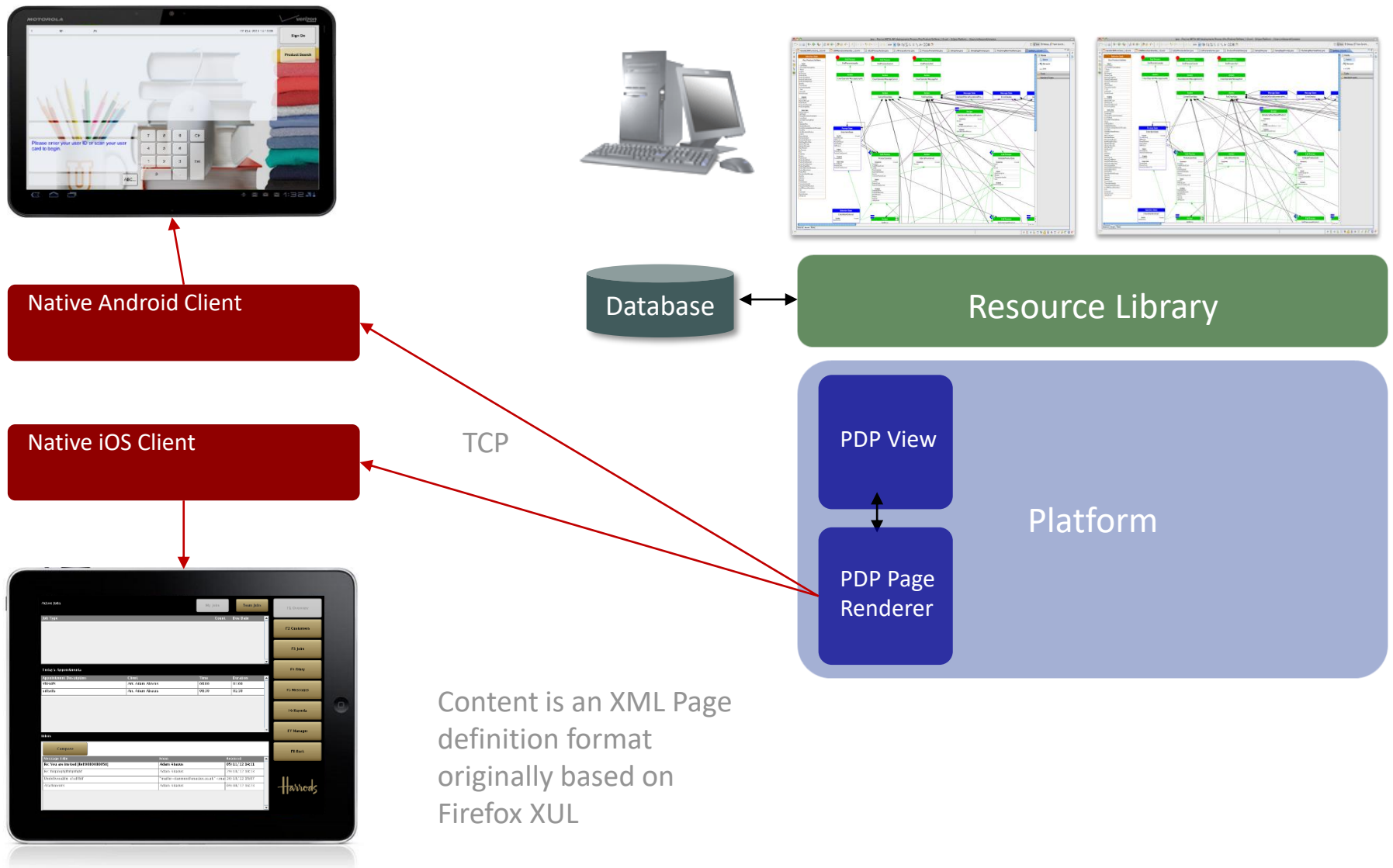
Kiosk Processes



Thick Client UI Architecture

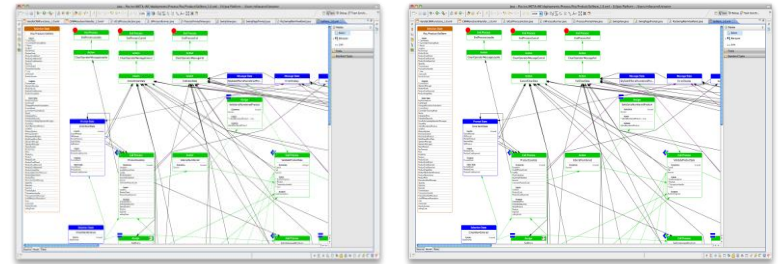
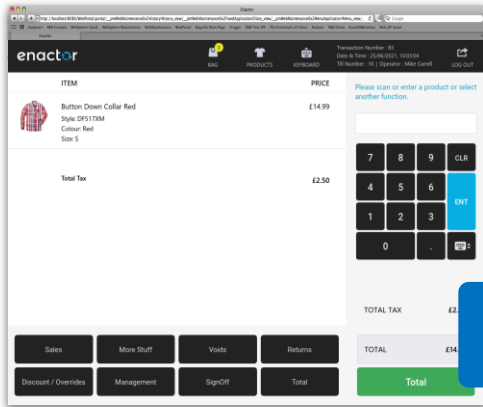


Android and IOS Client UI Architecture



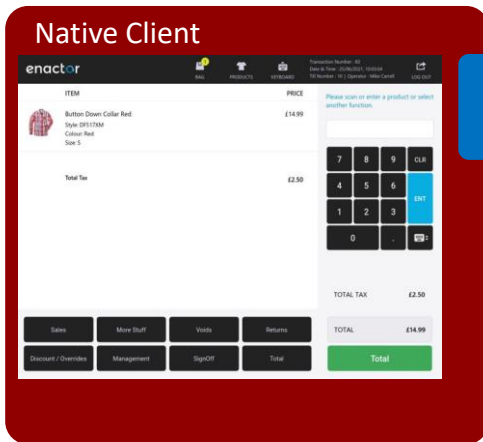
React/JavaScript UI Architecture

Plain Browser

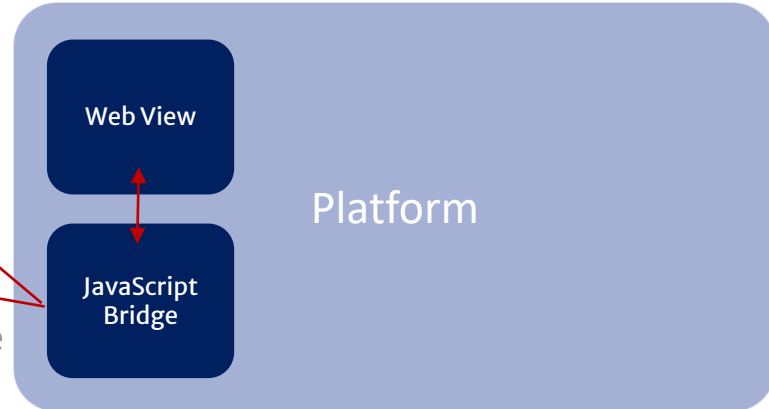


Web Connector

Embedded browser



Web Connector



Data is supplied to the UI as JSON

Embedded browser support with access to devices

 Seamless

 Integrated

 Consistent

 Personalised

enactor[®]
retail systems for a digital world

Q & A