

Neil J Martin
neilm@4js.com

Genero User Interface Course Overview

Genero BDL 3.10
Genero Studio 3.10



Goals

- Define key Genero terms and concepts
- Describe the Genero product suite
- Describe the Genero architecture
- Identify the usage of XML and DOM within Genero



Meaning

The name 'Genero' is derived from the Latin verb 'to create' or 'to produce'.

Genero - a product designed to improve creativity and productivity



Goals

Genero was developed with two main goals:

1. Support different display devices with the same source code.
2. Provide an abstract definition of the user interface that can be manipulated at runtime as a tree of user interface objects.



Overview

One business logic fits all...



Terms & Acronyms

Dynamic Virtual Machine (DVM) - Application/Runtime server, automatically manages and optimizes communication between clients and runtime system where the Business Logic and Design Logic are processed

Open Database Interface (ODI) - Part of the DVM, provides concurrent access to a broad range of databases

Abstract User Interface (AUI) - XML-based logical model of the user interface, which can be rendered at runtime to specific platforms

Dynamic User Interface (DUI) - DOM tree model of the AUI that can be modified at runtime



Components

Genero Compiler / Runtime – FGL

Core language compiler and runtime environment

Genero Web Services – GWS (supplied with the FGL package)

Communicate in real time with applications across heterogeneous environments using open standards based (XML, SOAP, WSDL ...).

Genero Application Server – GAS

Launch application and web services via a web server from a browser or the GDC.

Genero Studio – GST (bundled with other Genero components)

Integrated graphical set of tools designed to fast-track application development



Components

Genero Desktop Client – GDC

Desktop Client, One product fits Windows, Mac OS X, Linux.

Genero Browser Client – GBC (supplied with the FGL package)

Modern web client using (HTML5, JS, SASS, NODE)

Genero Mobile Android – GMA

Run application natively on Android devices

Genero Mobile iOS – GMI

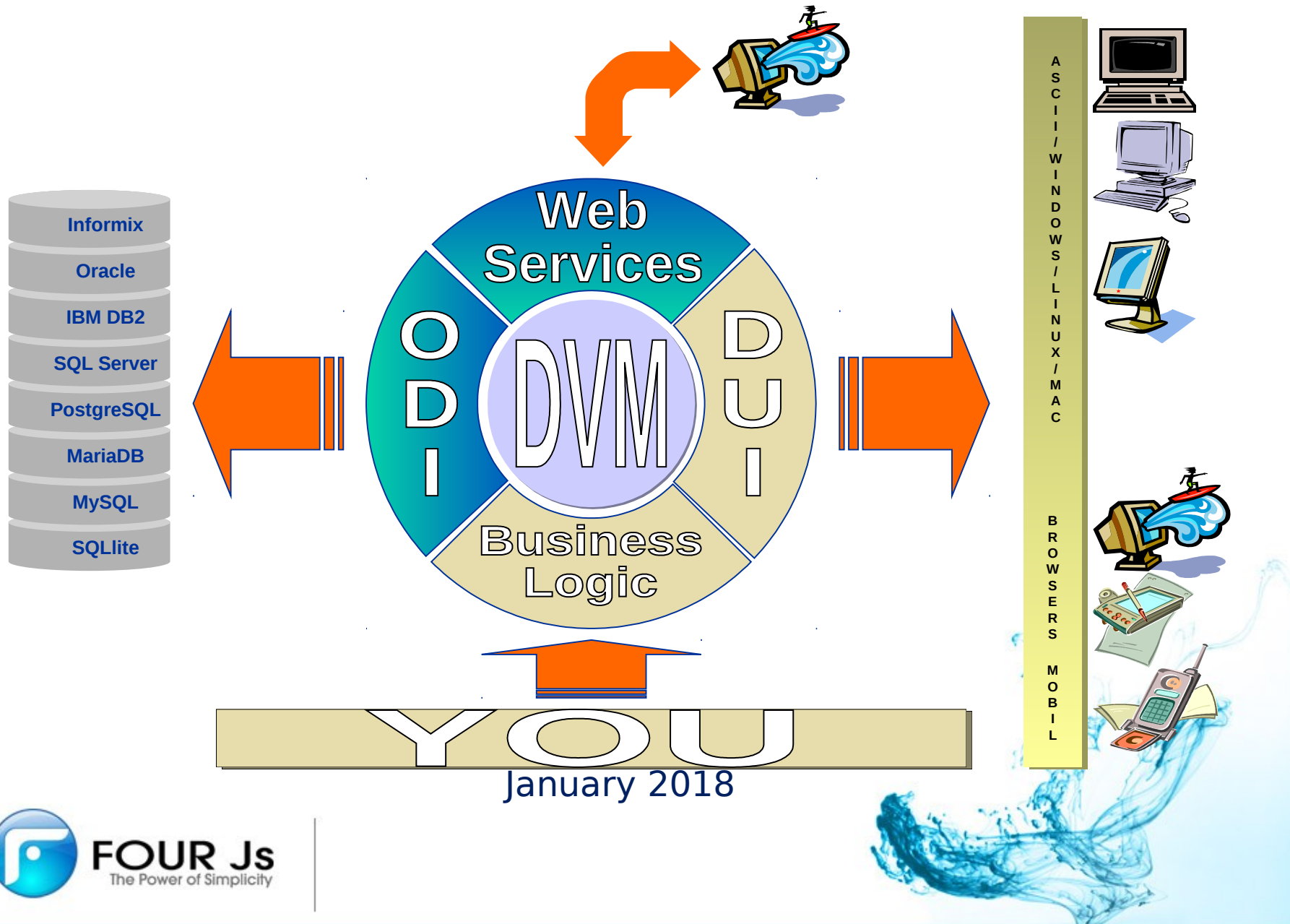
Run Genero application natively on iPhone/iPad

Genero Reporting Engine– GRE

Turn XML data into high level reports with multiple output choices.

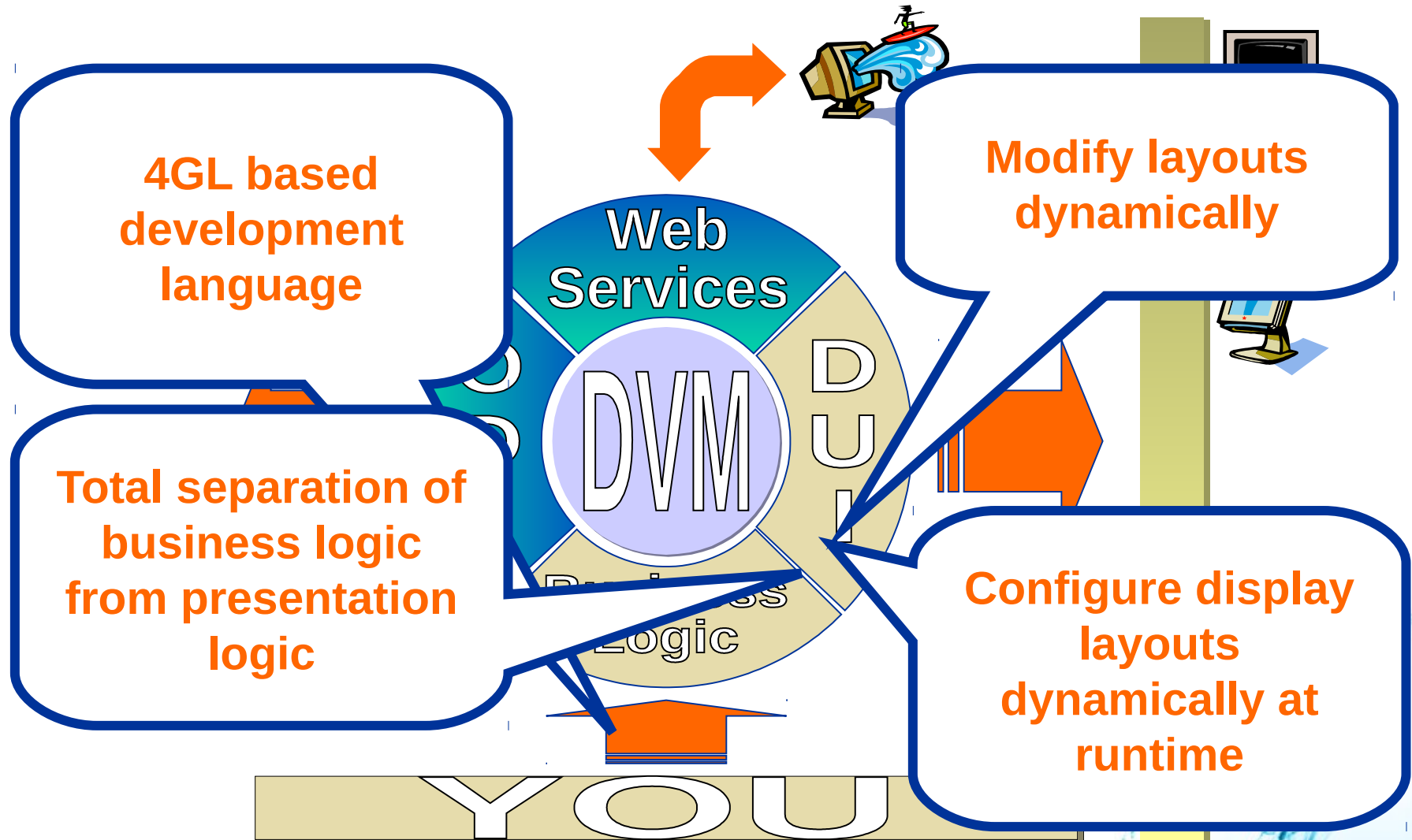


Genero Architecture

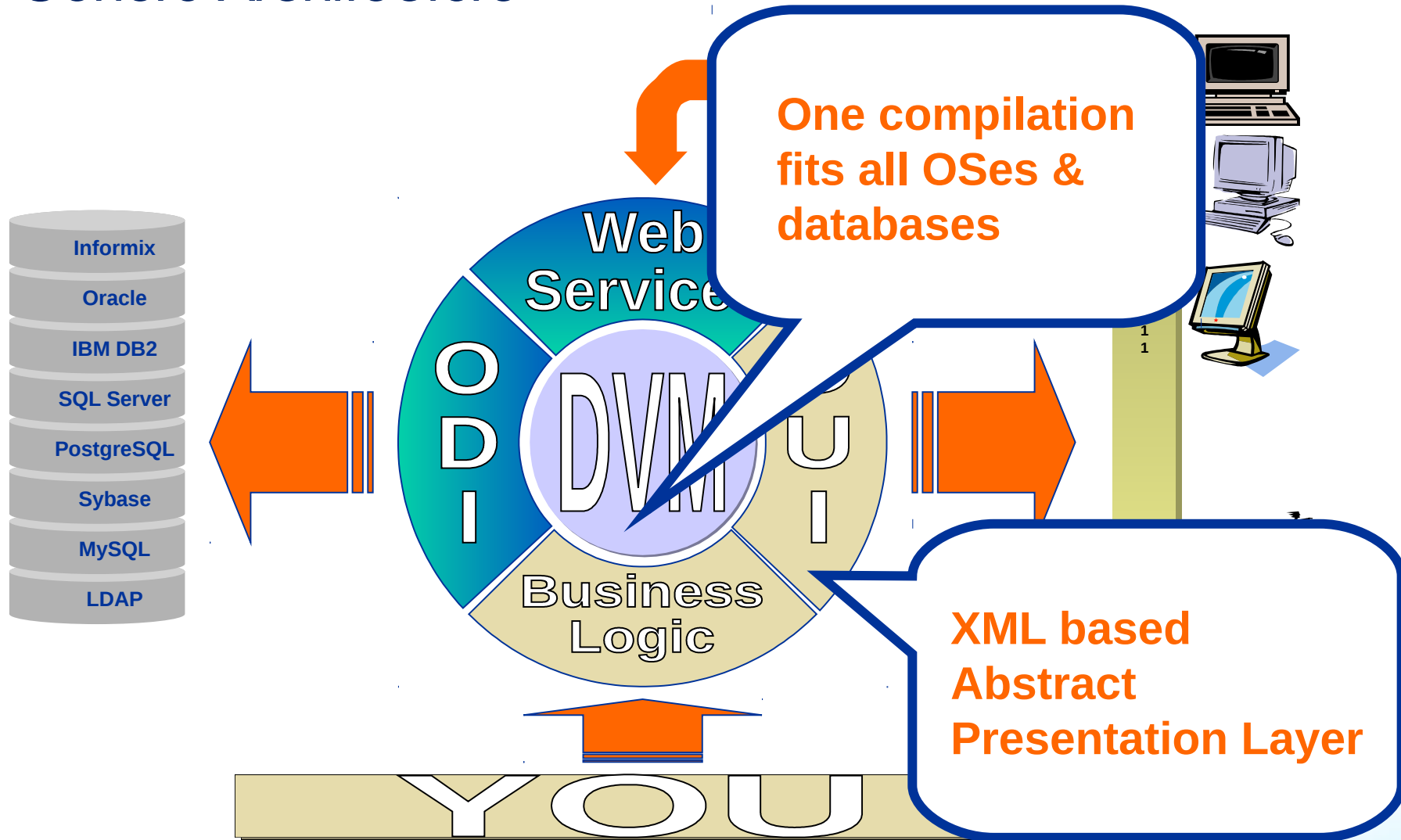


January 2018

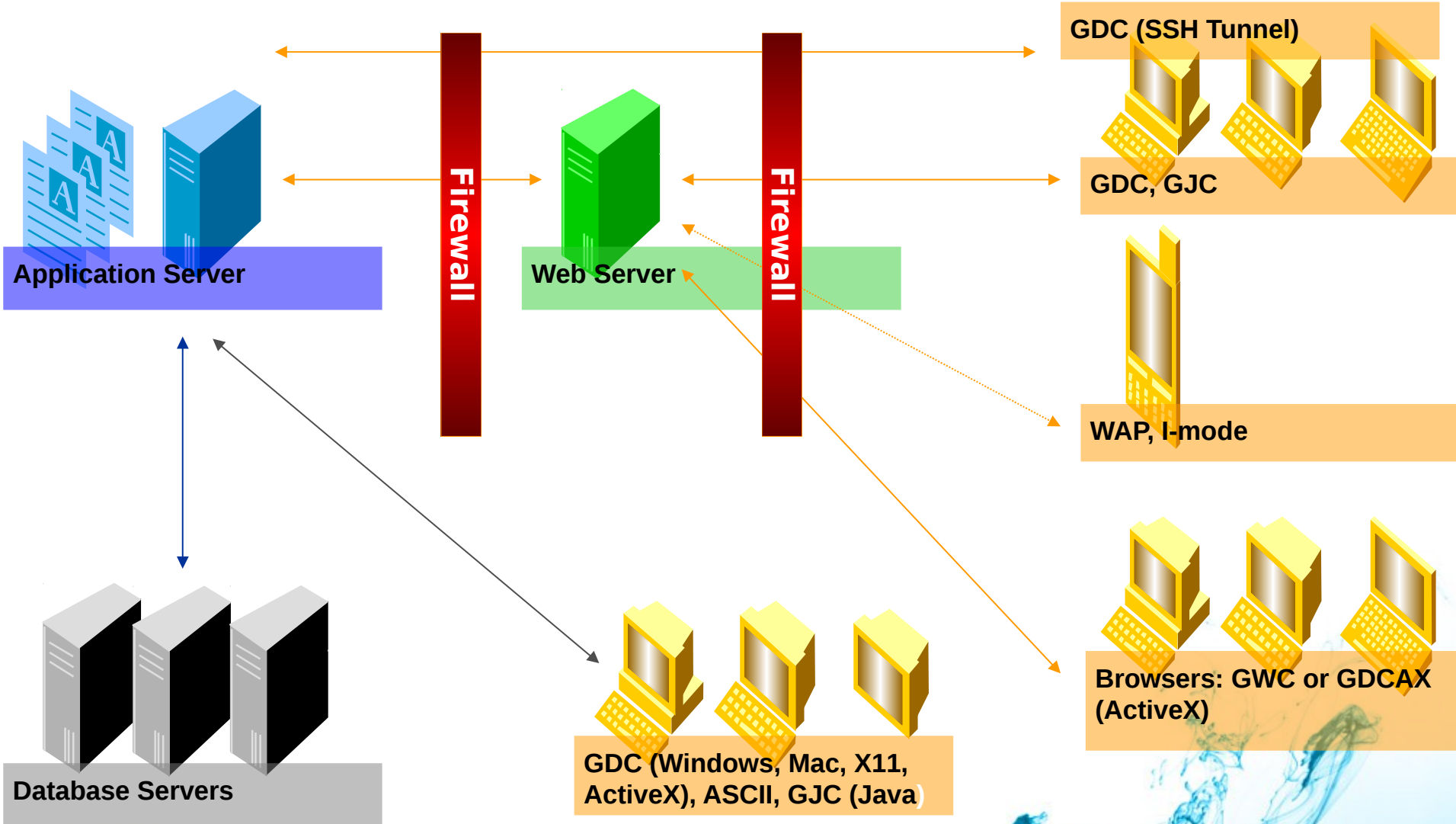
Genero Architecture



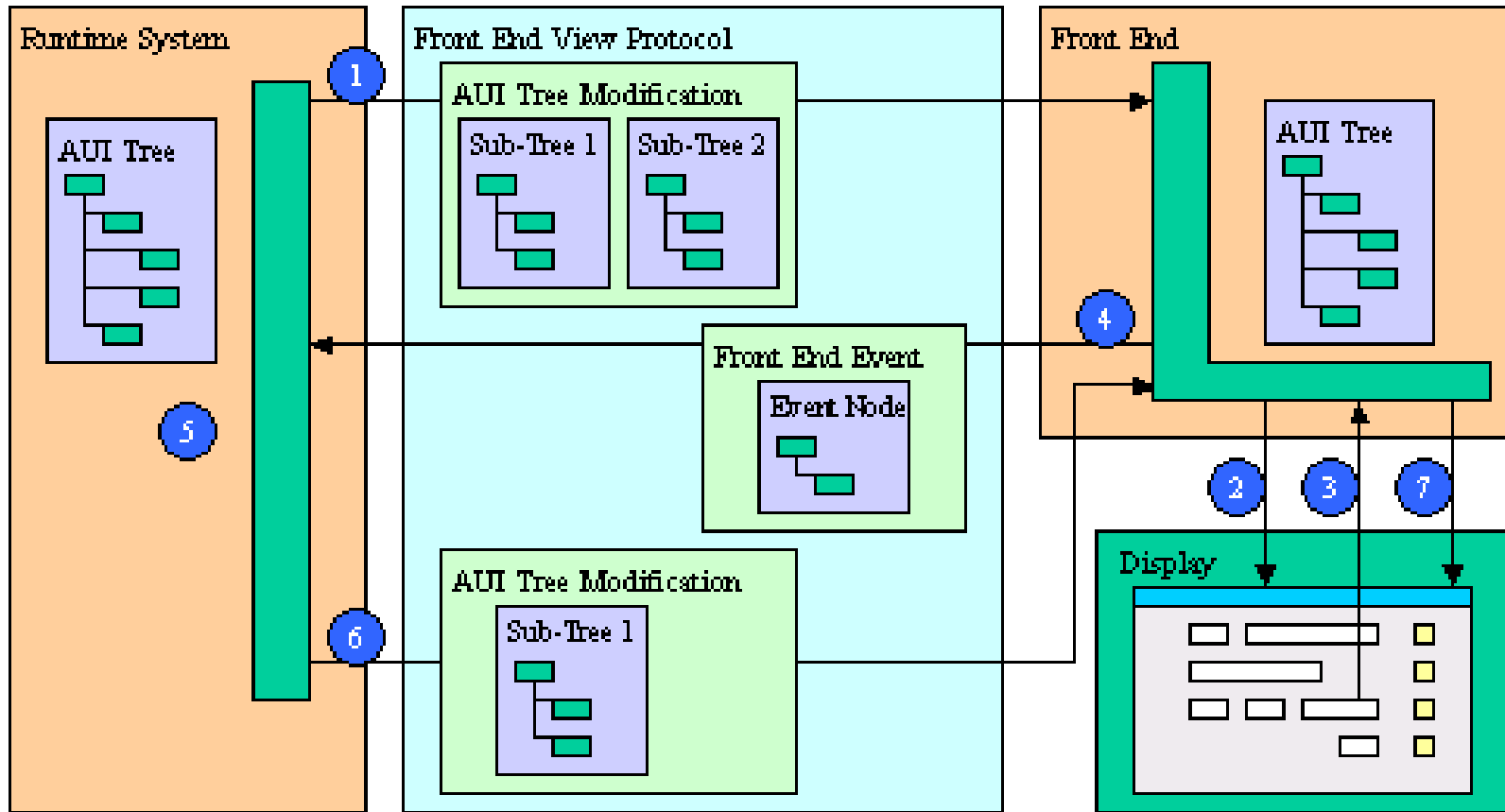
Genero Architecture



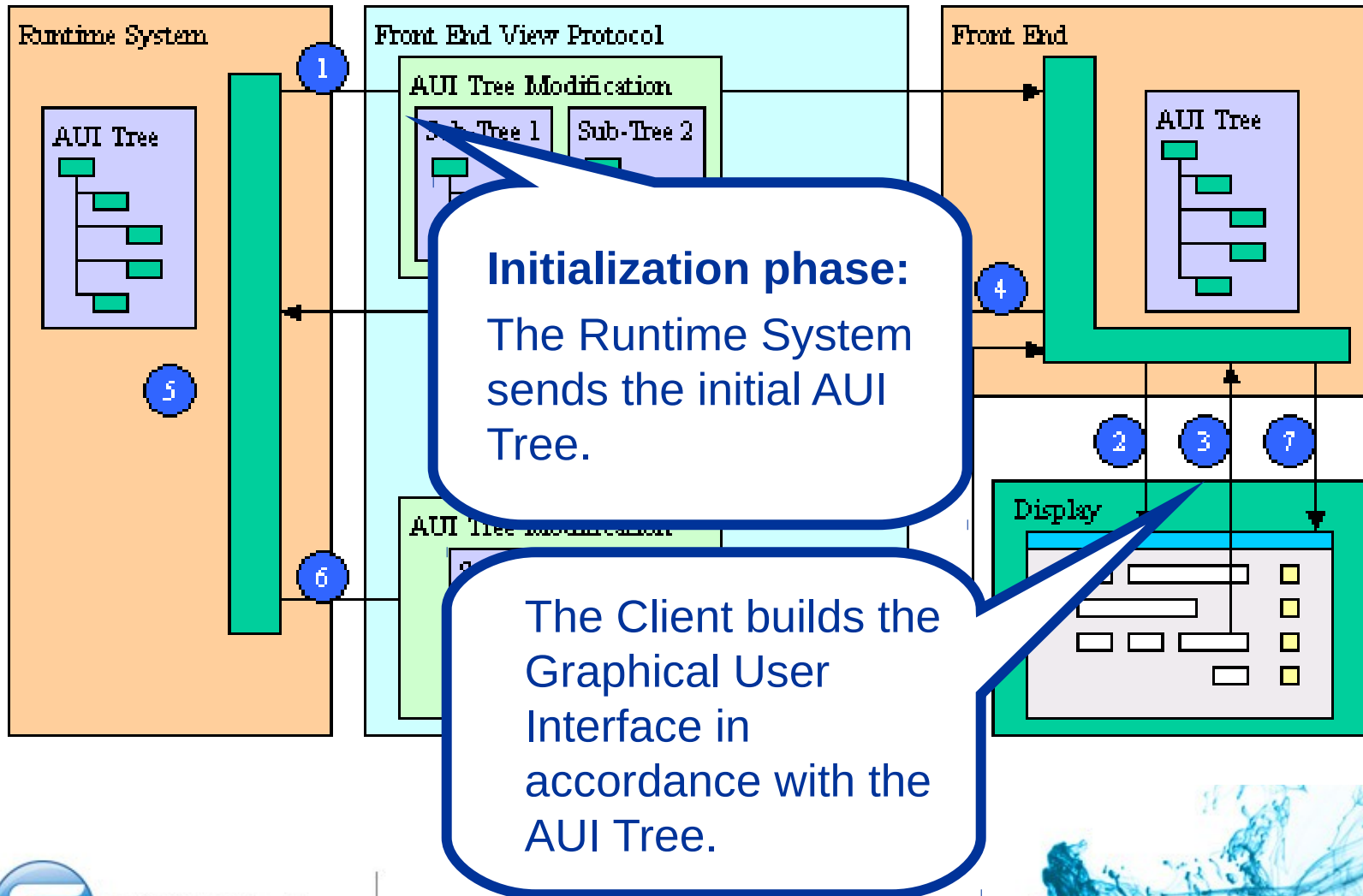
Genero Deployment



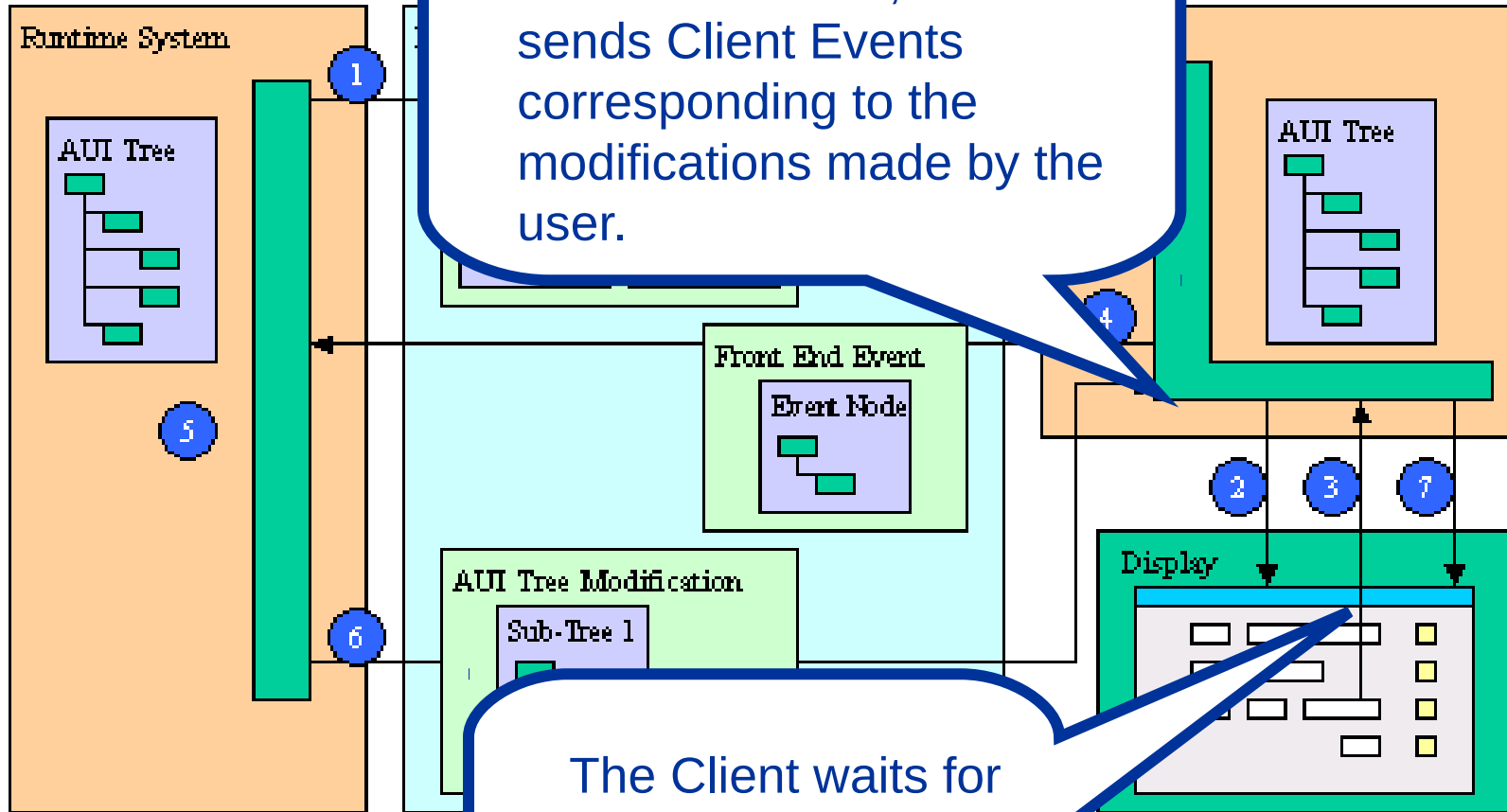
Genero Protocol



Genero Protocol



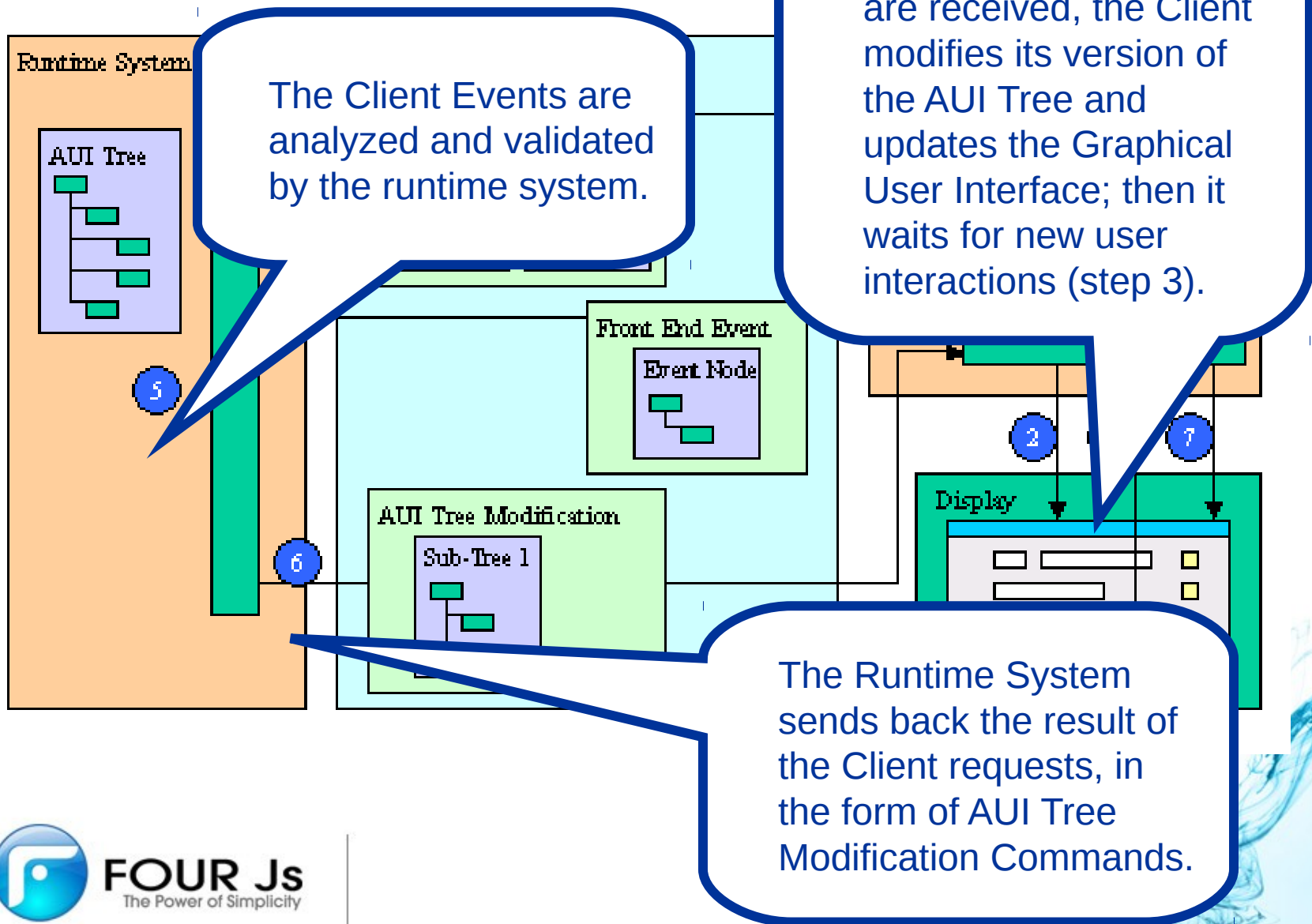
Genero Protocol



When the user performs some interaction, the Client sends Client Events corresponding to the modifications made by the user.

The Client waits for a user interaction (mouse click, keyboard typing).

Genero Protocol



XML

XML = eXtensible Markup Language

- A meta-language that allows the design of a markup language, used for the easy interchange of data over the internet.
- XML is simply tagged text - it does not do anything on its own.
- XML tags are not predefined. Tags are defined by the designer.

Genero can read, write, and interpret XML documents.



XML

Tags

Elements (root, child)

```
<Form name="sampleform" width="23" height="4">  
  <Grid width="23" height="4">  
    <Label name="label1" width="20" text="Hello"  
      posY="1" posX="2" gridWidth="20"/>  
    <Label name="label2" width="20" text="Dolly"  
      posY="2" posX="2" gridWidth="20"/>  
    ...  
  </Grid>  
</Form>
```



XML → DOM

Document Object Model

- The DOM is an internal tree representation of an XML document.
- The DOM tree resides in memory.

An XML document with the extension .42f is automatically generated when a Genero form definition file (.per) is compiled.

The internal representation of this XML document is our AUI DOM tree.



Summary

Genero is

- A N-tier architecture
- Supports different display devices with the same source code
- One compilation fits all OSes and databases
- Enables dynamic local & global display layout modifications
- Uses XML and DOM tree to represent the user interface
- Includes built-in classes to manipulate the user interface elements of the DOM tree



Q&A



Intelligent Business Application Infrastructure