
JavaOne™

Building HTML5 Web Apps with Avatar (CON7042)

Santiago Pericas-Geertsen
Torkel Dominique
Sumathi Gopalakrishnan

Program Agenda

- What is Avatar?
- Syntax, syntax and more syntax
- Avatar by example
- Advanced concepts
- Conclusion

What is Avatar?

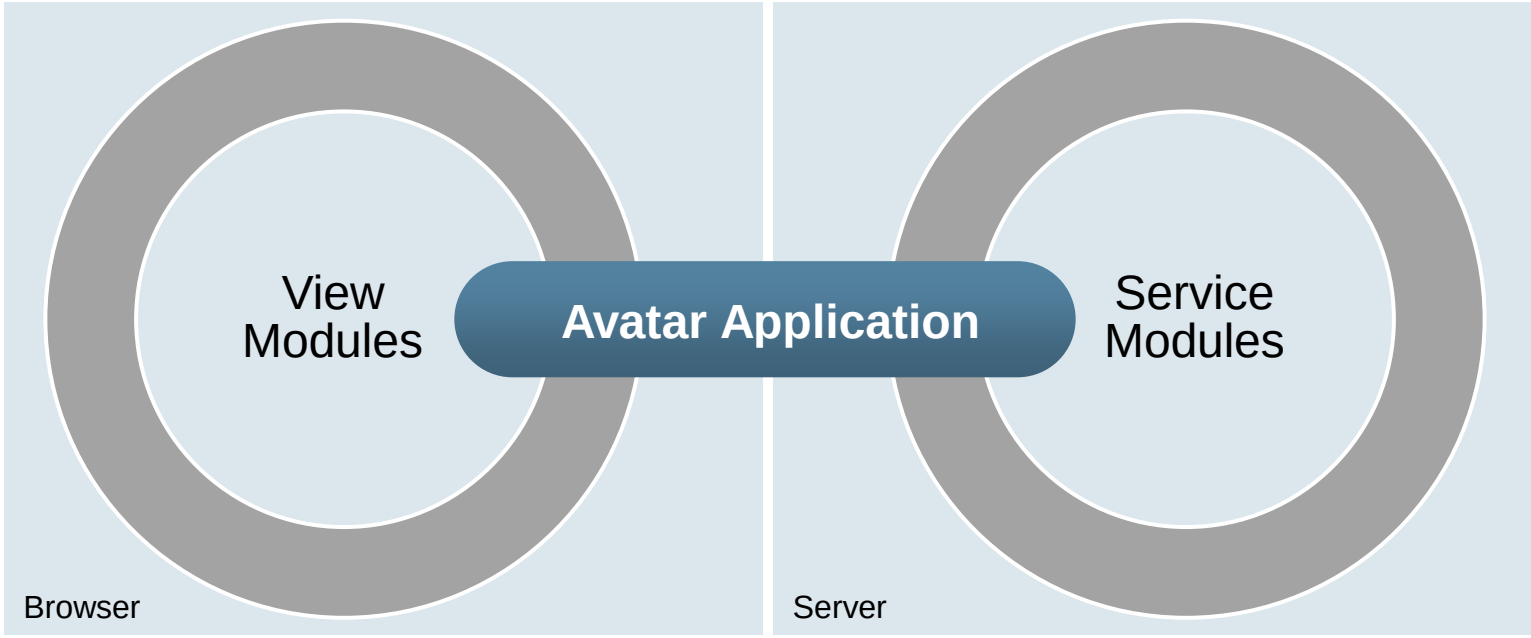
What is Avatar?

Avatar in a nutshell

Avatar is a modular, end-to-end web development framework for building enterprise mobile and desktop applications using JavaScript, HTML5 and a thin-server architecture.

Modular and End-to-End

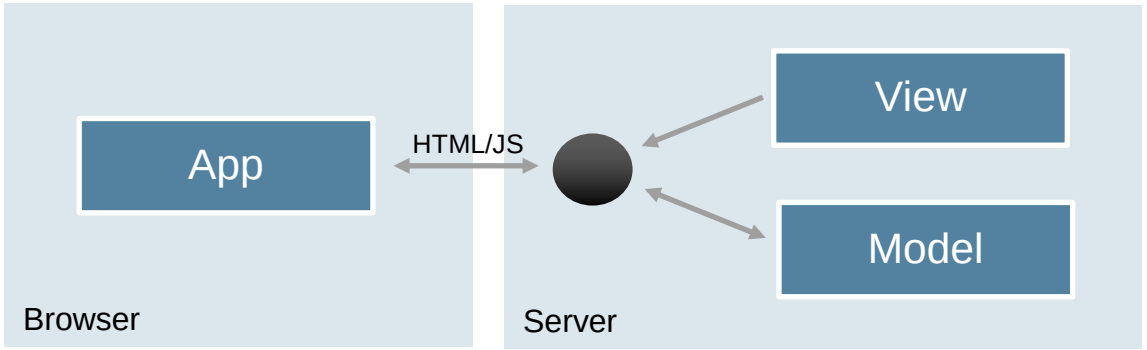
Avatar Application Structure



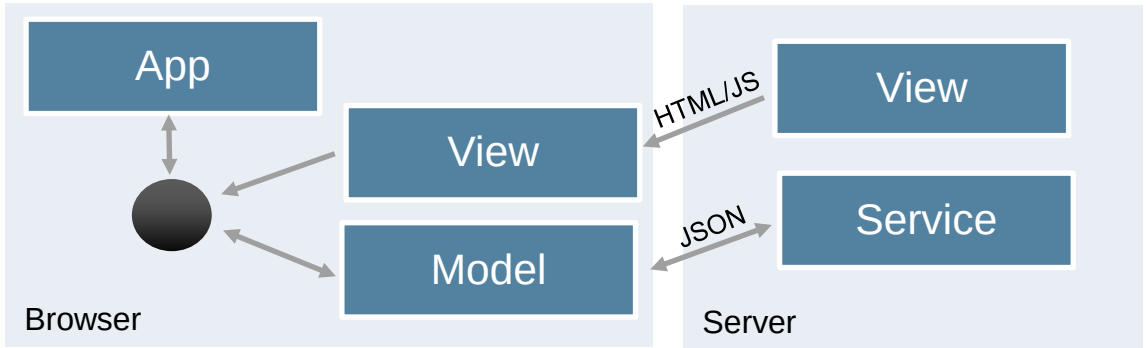
Thin-Server Architecture (TSA)

Traditional vs. Thin Server

Traditional



Thin Server



Programming Model

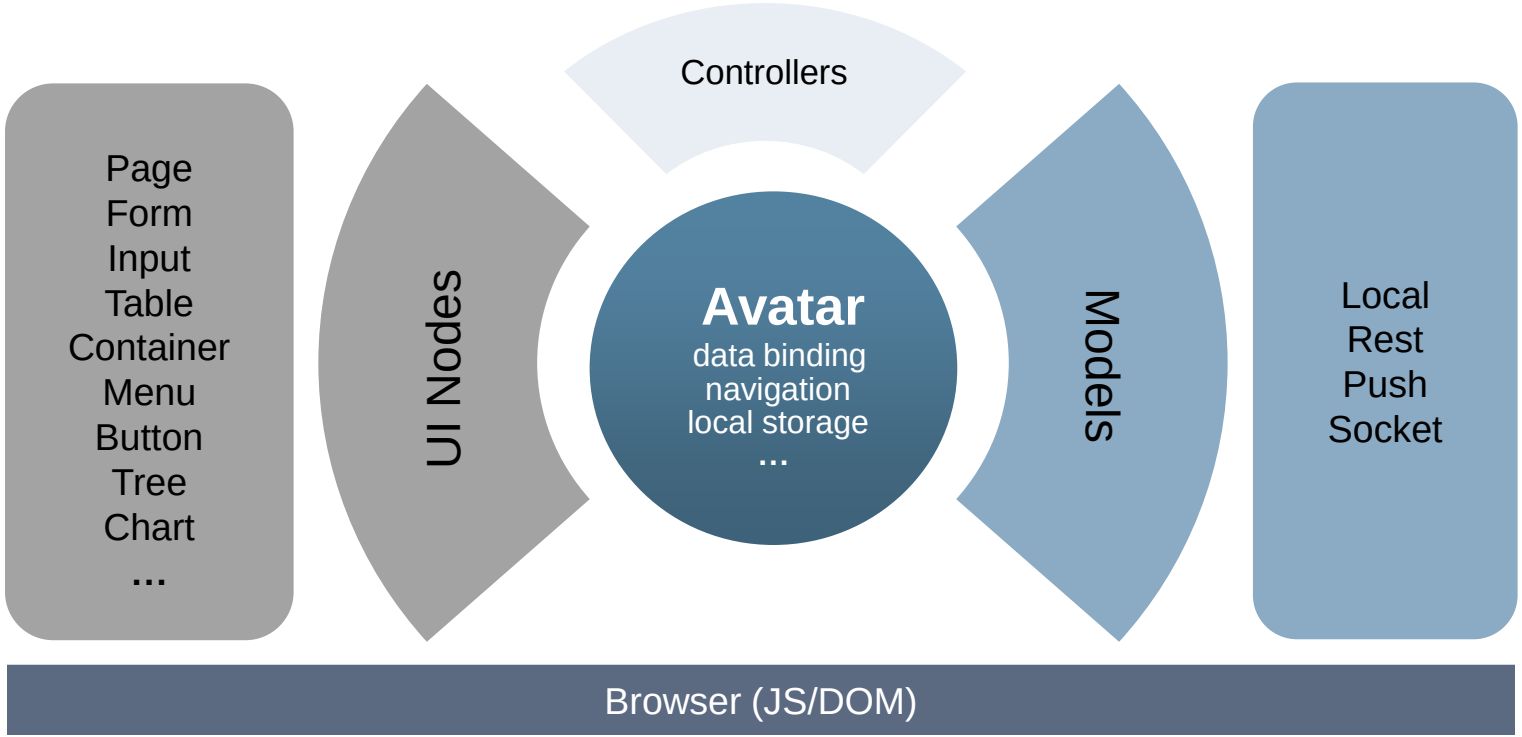
Client and Server

Client	Server
• Extensible component views • Pluggable widget sets • Models in JavaScript • Data binding • HTML and CSS support	• Java EE container ➢ JAX-RS, WS (JSR-356), … • Avatar services in JavaScript ➢ REST, SSE, and WS ➢ Nashorn JavaScript engine ➢ Access to Java API

Avatar Module System (AMD)

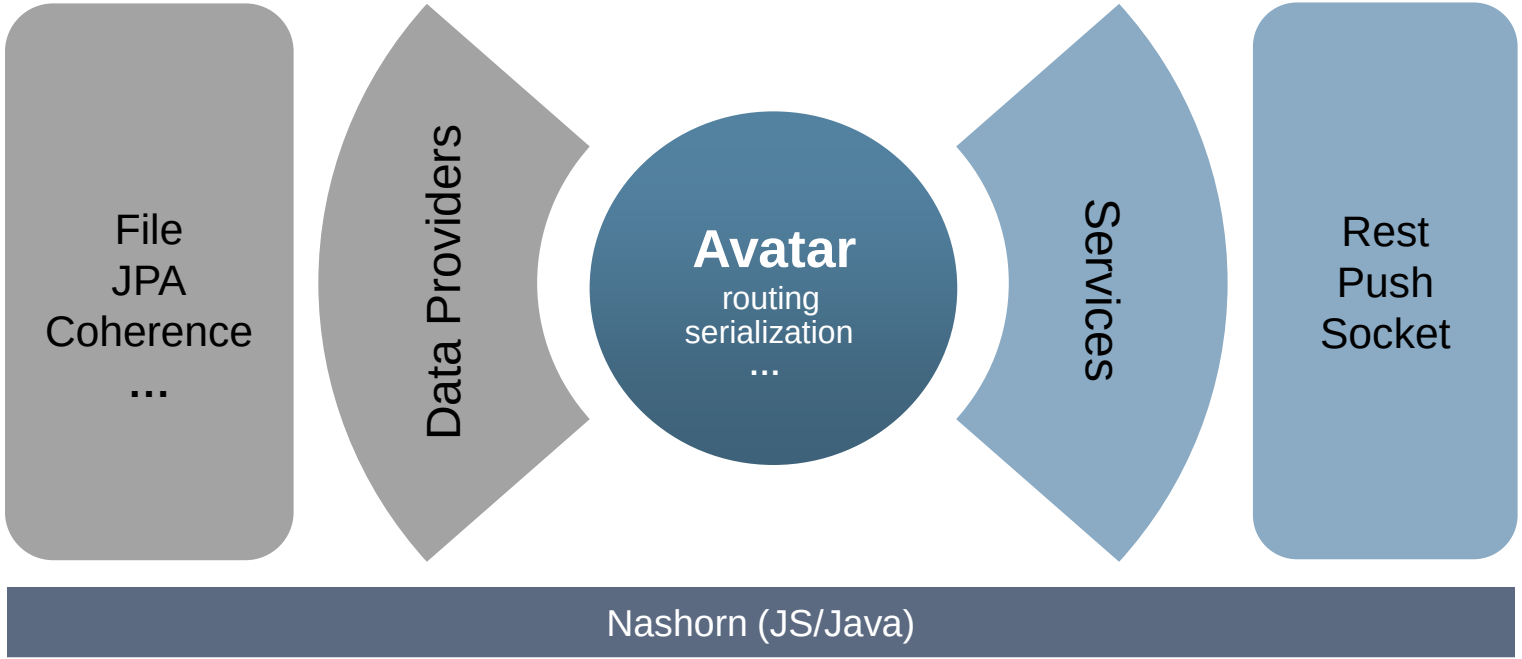
Avatar Client

Concepts



Avatar Server

Concepts



Syntax, Syntax and more Syntax

Programming Style

Declarative vs. Imperative

Declarative	Imperative
• Good for views (UI) ➢ Containment by nesting • Data binding using EL • Templating • Tooling	• Good for models and controllers • WYSIWYG • Full JavaScript control

What's the Syntax?

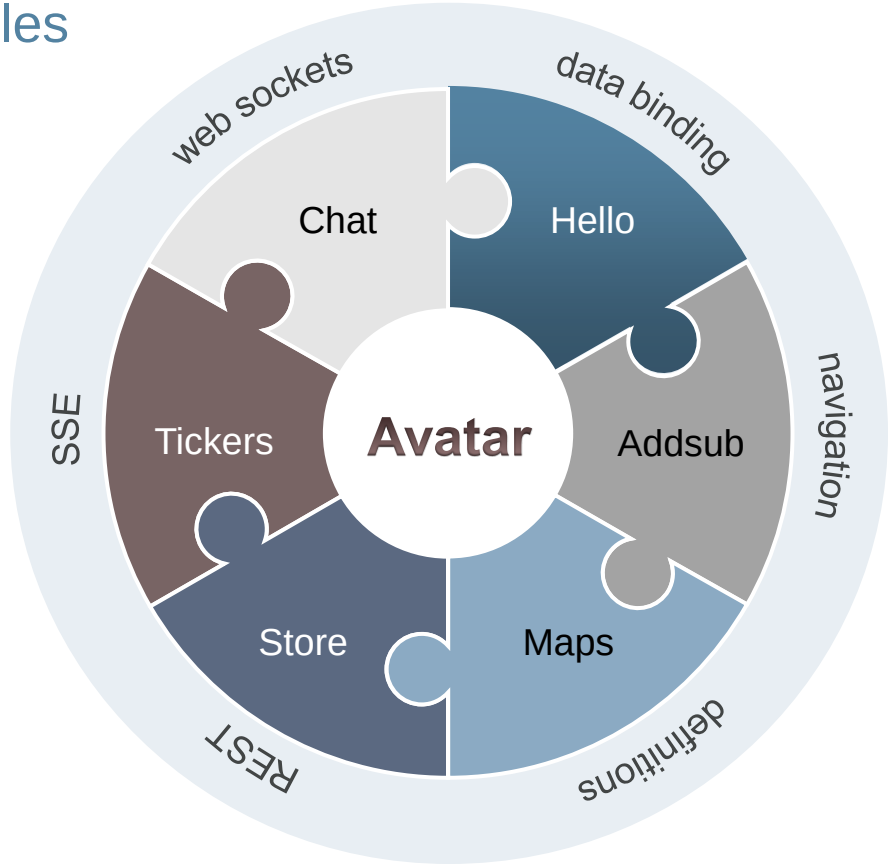
XML vs. HTML

XML	HTML
• "X" is for extensible ➢ Support for namespaces • Components ➢ Easily defined and used • HTML embedding	• We all know it • What the browser understands • Less structured • Extensions via "data-" ➢ No namespaces
Supported	*Under investigation*

Avatar by Example

Avatar by Example

Simple Examples



Where's Hello World?

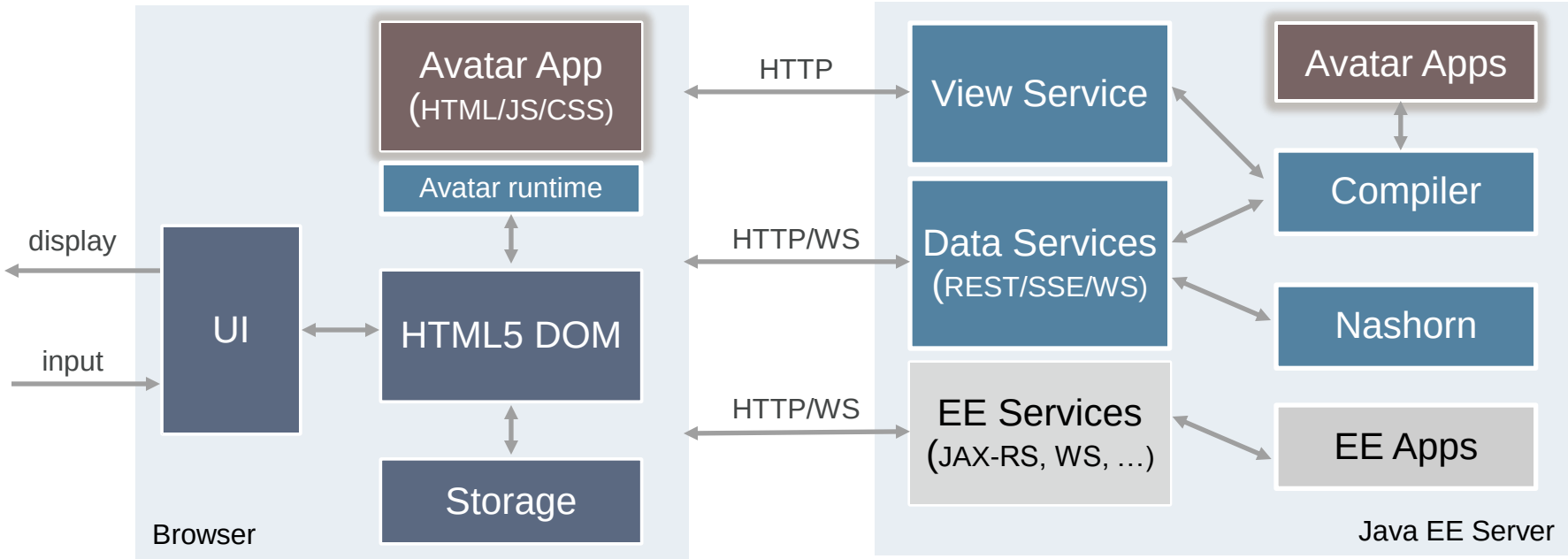
Single View Module

```
<viewModule … >
  <localModel>
    var NameModel = function() {
      this.first = "Planet";
      this.last = "Earth";
      this.clear = function() {
        this.first = this.last = "";
      }; };
  </localModel>
```

```
<page id="home" title="Hello World">
  <model id="name" idref="NameModel"/>
  <form>
    <input label="First Name" value="#{name.first}"/>
    <input label="Last Name" value="#{name.last}"/>
    <output value="Hello #{name.first} #{name.last}”/>
    <button label="Clear" action="#{name.clear()}“/>
  </form>
</page> </viewModule>
```

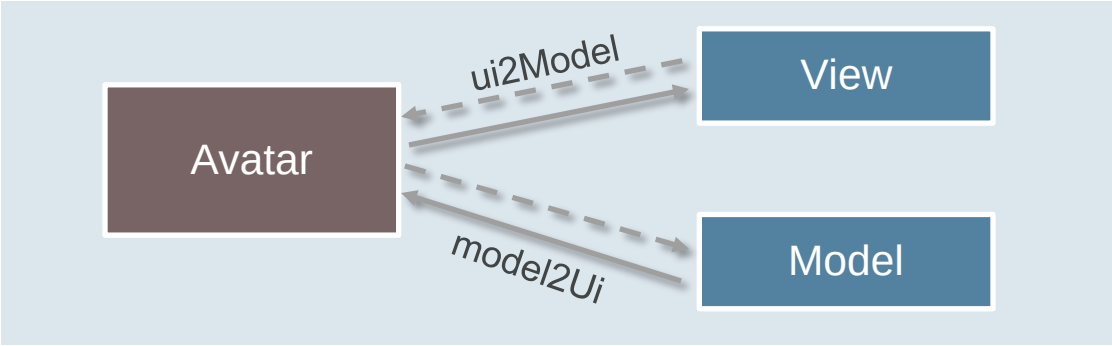
Avatar Runtime

Architecture



Data Binding

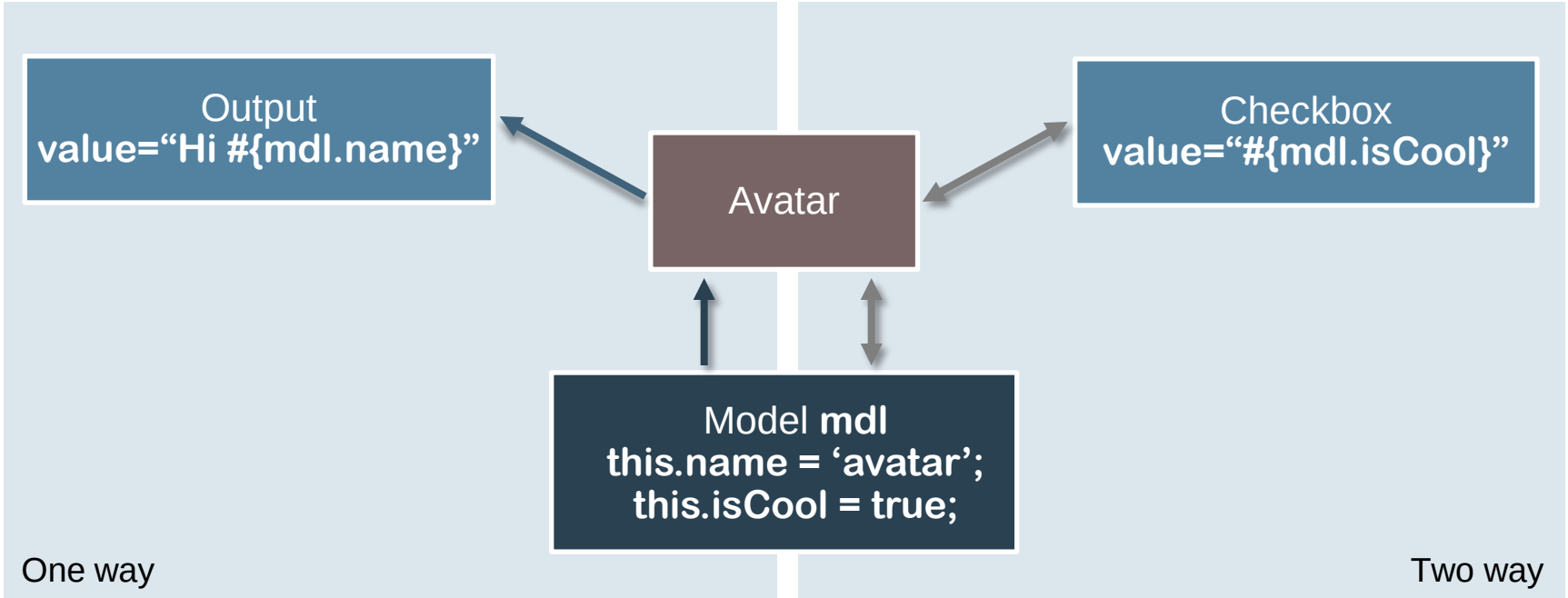
One way or two way



UI Node Property Types	HTML Attributes
➢ **One way**: Model to UI only ➢ **Two way**: Binding for input-type properties ➢ **Action**: No binding, for actions with side effects	➢ **One Way**: Non-action attributes ➢ **Action**: Attributes like **onclick**

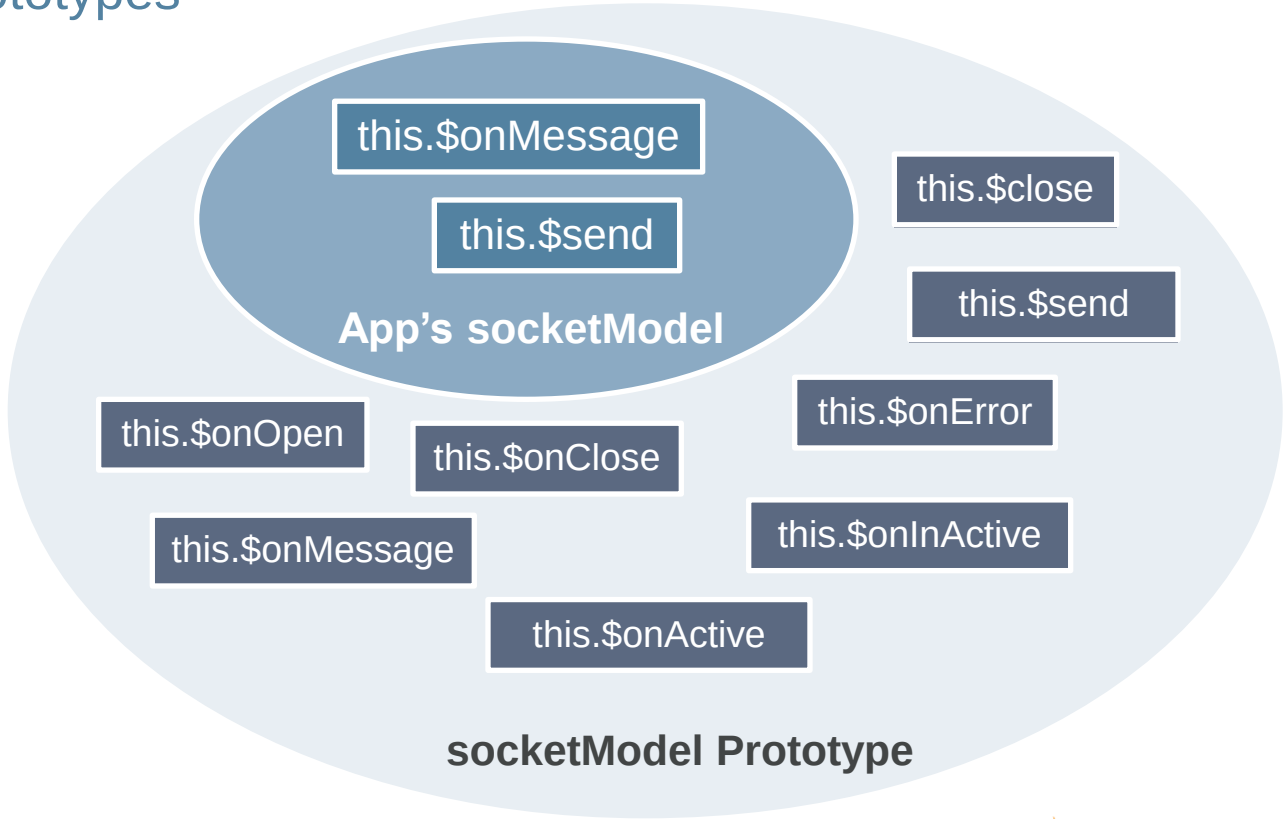
Data Binding

Example



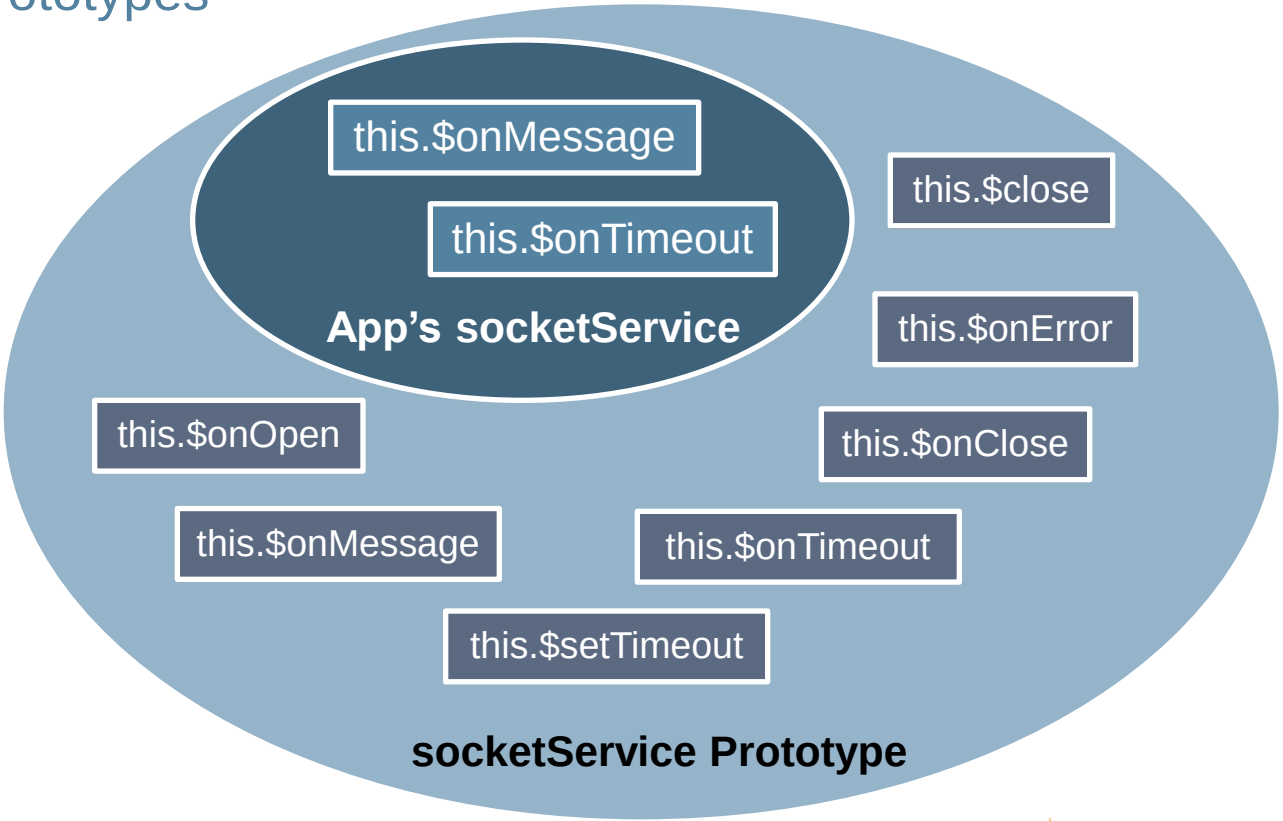
Models and Services

Model Prototypes



Models and Services

Service Prototypes



Models and Services

Chat Example

```
<socketModel>
  var ChatModel = function() {
      this.message = “”; this.user = “”;
      var superSend = this.$send;
      this.$send = function() {
          this.process(this.message);
          superSend.call(this);
      };
  … };
</socketModel>
```

```
<socketService url=“websockets/chat”>
  var SocketService = function() {
      this.$onMessage = function(peer, message) {
          this.process(message);
          peer.getContext().sendAll(message);
      };
  … };
</socketService>
```

Navigation in Avatar

Hash Scheme

 # <page-id> / <pane-id> ? <parameters>

- ID of page to navigate
- May contain module name if in other module

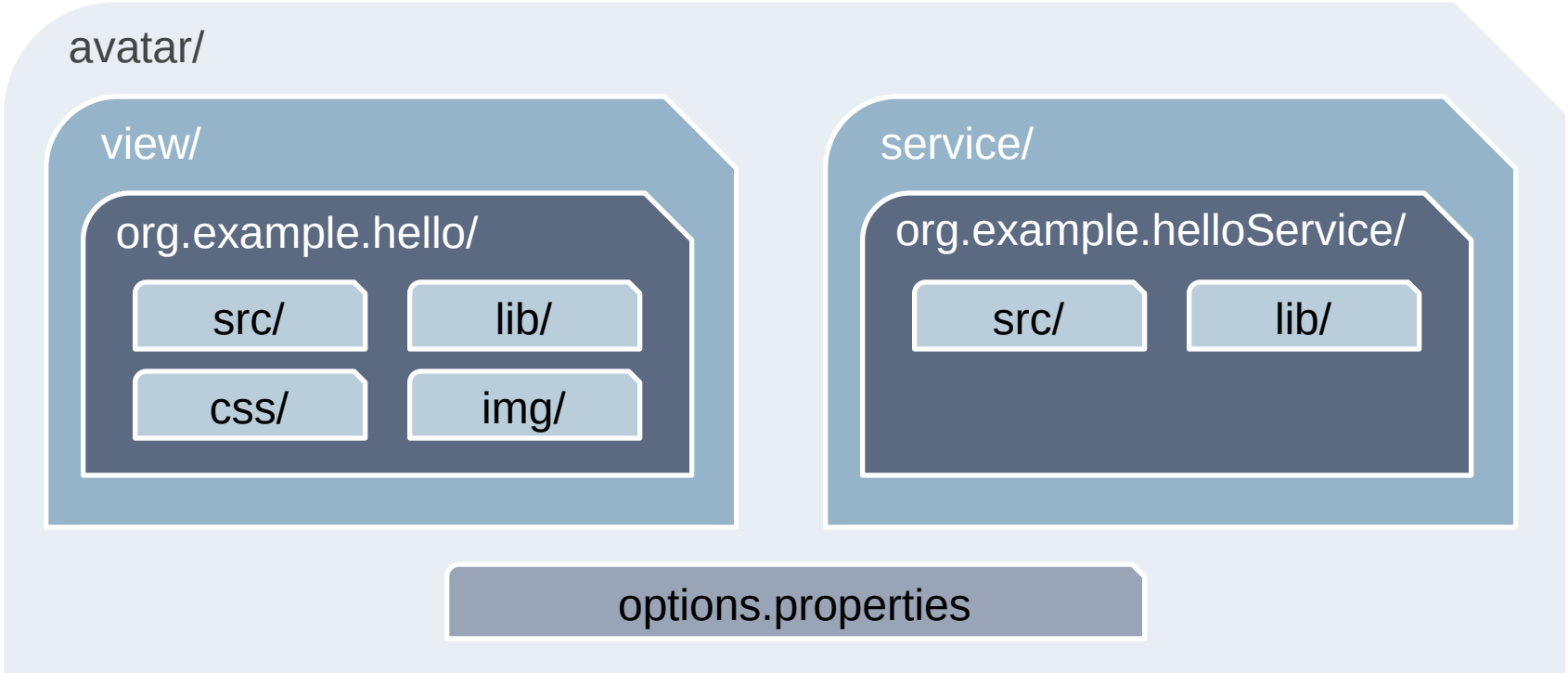
- ID of pane in page
- Selects pane in stack or tab container

- Same syntax as URL parameters
- E.g.: n1=v1&n2=v2

Global model **$location** available in EL

Avatar Archive

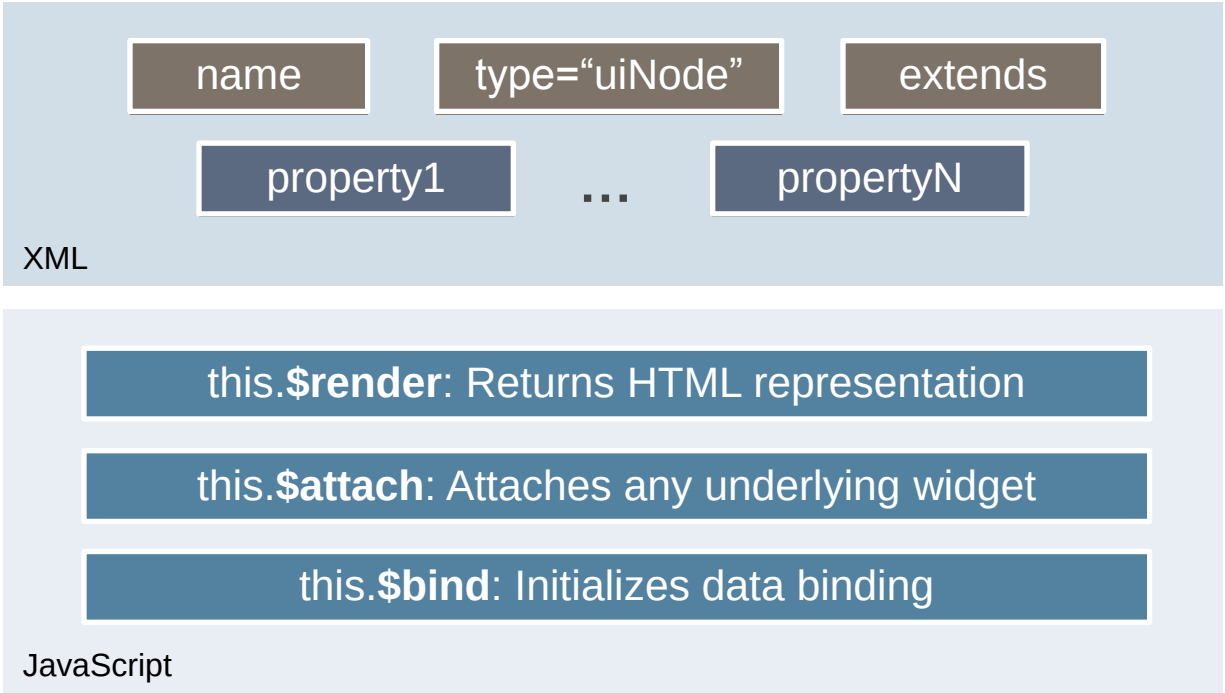
A ZIP file



Advanced Concepts

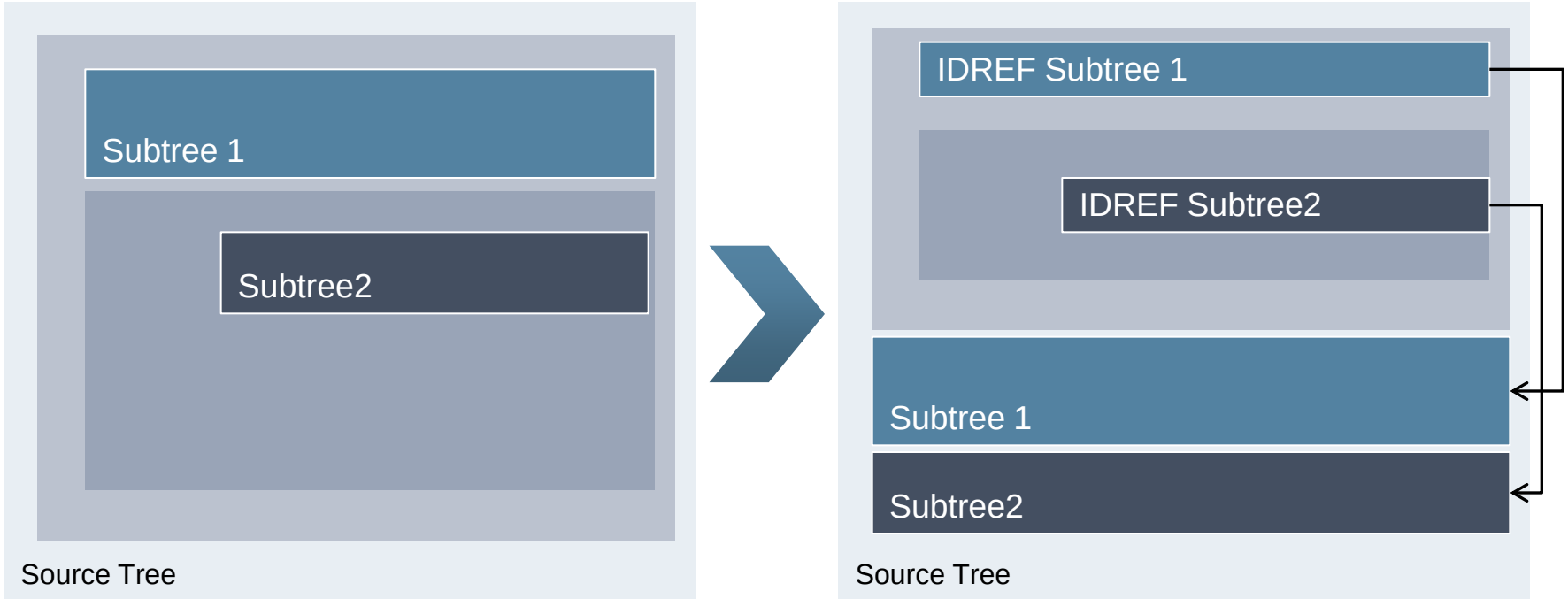
Advanced Concepts

UI Nodes Defintions



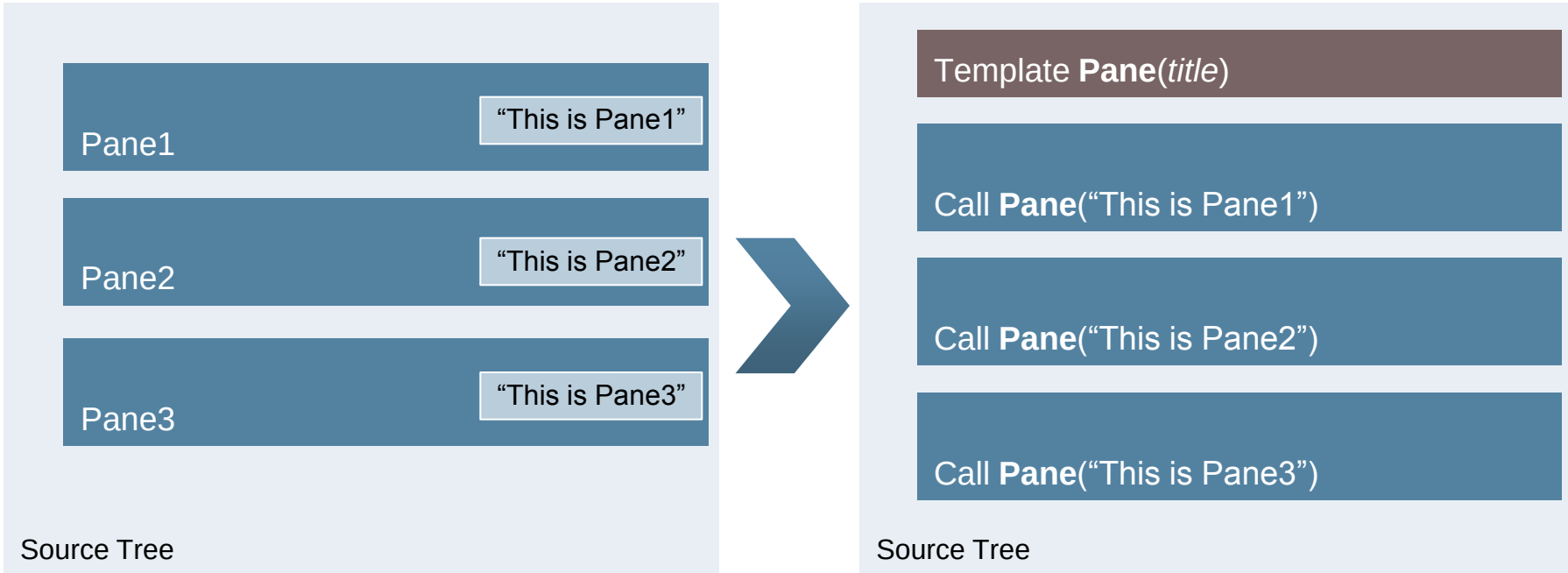
Advanced Concepts

Tree References



Advanced Concepts

Templates



Advanced Concepts

Like and Extensions

Like	Extensions
• Inherit properties from another node instance • Avoids duplication of attributes	• Simple tree transformations • Insert, replace, remove, … • Customize core application

Conclusion

Conclusion

Wrapping up

What is Avatar?

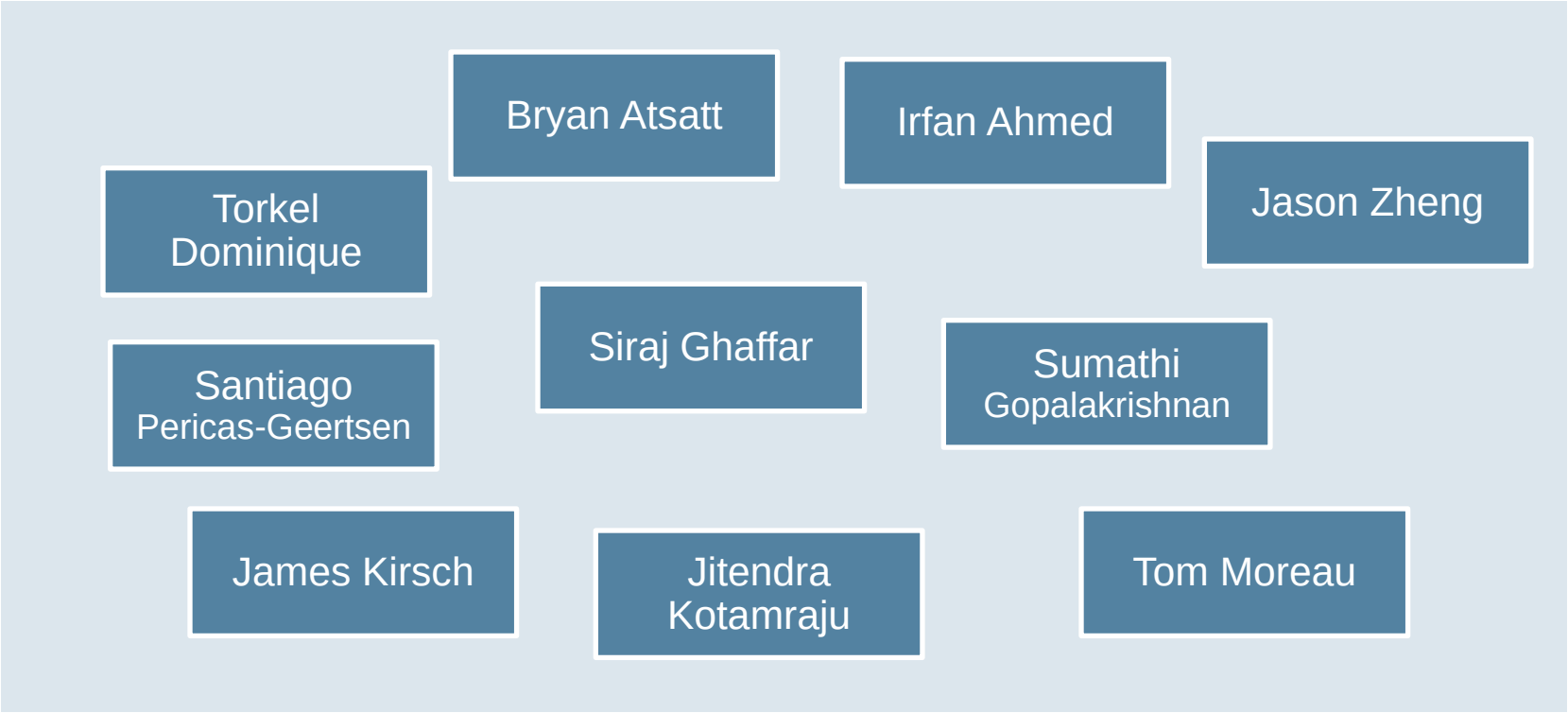
- Web framework for building TSA apps
- End-to-end, extensible and modular
- HTML5 features
- Suitable for small and large projects
- Runs on Java EE

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.


JavaOne™

Avatar Team

Meet the team



One More Demo

Real World Application