instantiations

Advancing Modern Web Application Architecture in Seaside

Seaside + Hotwire + Websockets

Johan Brichau

VAST Consultant and Senior Software Engineer

jbrichau@instantiations.com

Agenda

instantiations

Instantiations is investing in ongoing Seaside development. As a result, Websocket features are available in Pharo first. They are scheduled for later release in VAST 2026.

seaside

instantiations

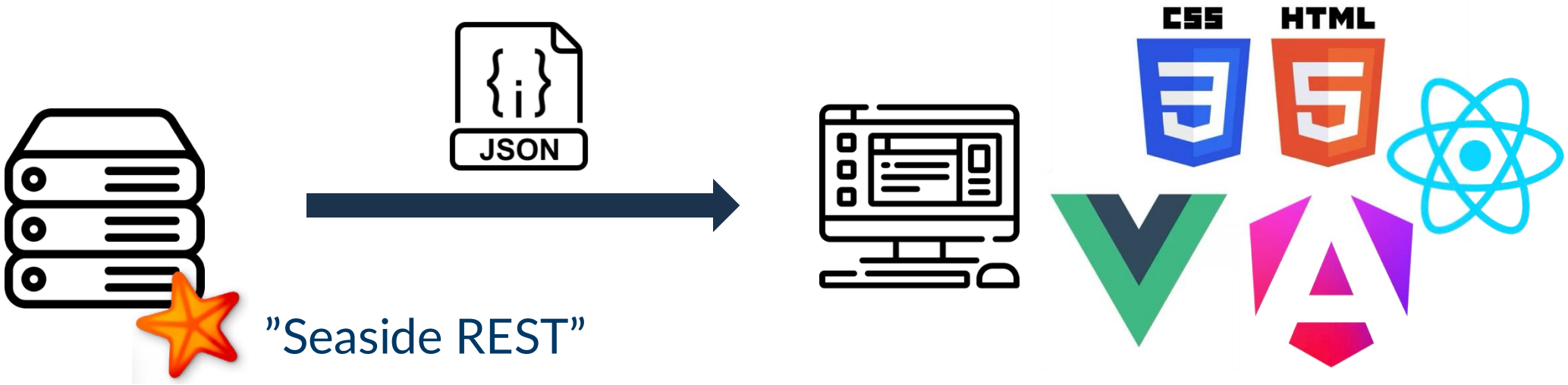
Building web applications with seaside

- Server-side web framework
- Natural code flow
- Reusable, stateful components
- Programmatic HTML generation
- jQuery integration
- Support in VAST, Pharo, GemStone

“Classic” Multi-Page WebApp Architecture

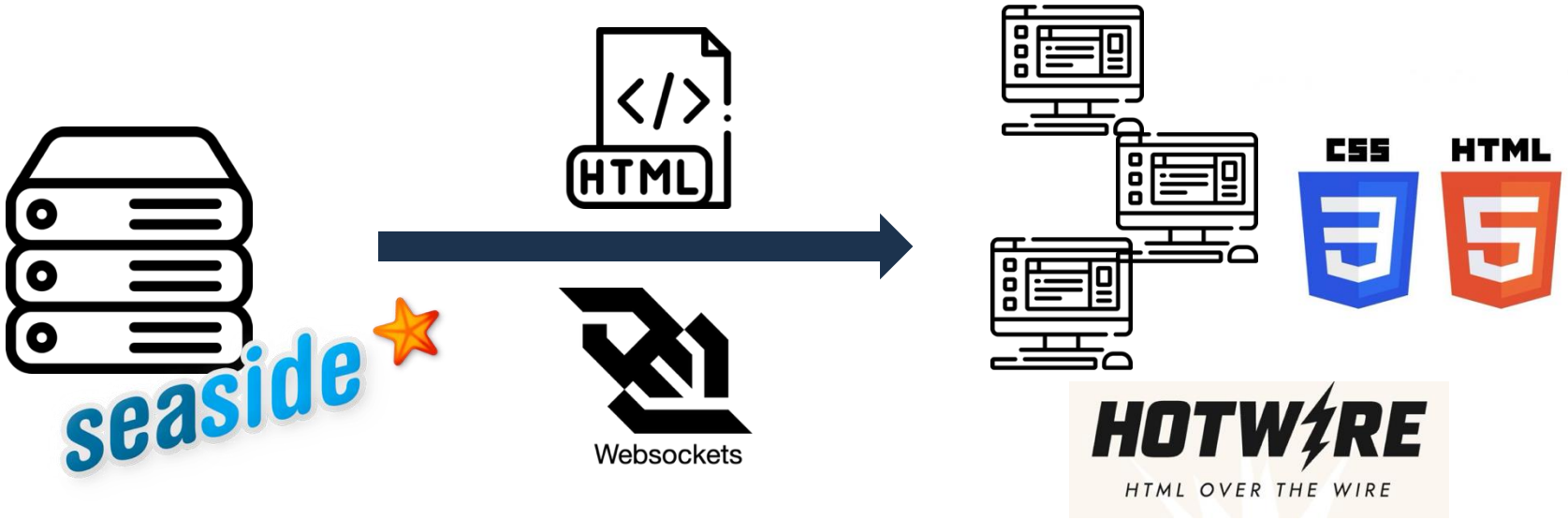
- “Simple” Architecture
- All logic on the server
- **Full page navigation**: webapp is set of linked webpages

“Common” Single-Page WebApp Architecture



- Better user experience with partial page updates
- **Complex architecture**
- Application logic divided and replicated in client and server

This presentation:
Server-side-rendered Single Page WebApp



- “Simple” Architecture
- All logic on the server
- Better user experience with partial and live page updates

HOTWIRE

HTML OVER THE WIRE

instantiations

What is Hotwire?

- Client-side library to augment server-side rendered pages
  - Declarative html attributes to drive the logic
- Implement Single Page behaviour from the server
- Origins in Ruby-on-Rails but server-framework agnostic
- **Turbo**
  - Partial page updates
- **Stimulus**
  - Javascript controllers attached to page elements

HOTWIRE

HTML OVER THE WIRE

https://hotwired.dev/

Seaside + Hotwire

- Turbo frames are independent pieces of a web page that can be updated independently.
- Links inside a frame only trigger an update of that frame, by default.
- Turbo frames in Seaside are implemented using component decorations.
- Use regular **`callback:`** and **`call:`** / **`show:`**
  - Convenience **`turboShow: / turboCall:`**
- Seaside is optimised to render only the requested turbo frame
- Turbo streams deliver page changes as Replace / Update / Remove / Append / Prepend / Before / After of page elements
- **`turboStream:`** ajax-style callbacks in Seaside

WATurboFrame decoration

http://localhost:8080/examples/turbo/multicounter

Method: WATurboMultiCounter>>initialize

Seaside-HotwireTurbo-E
Seaside-InternetExplore
Seaside-JSON-Core
Seaside-Pharo-Canvas
Seaside-Pharo-Continua
Seaside-Pharo-Core
Seaside-Pharo-Developr
Seaside-Pharo-Email
Seaside-Pharo-Environm
Filter...

WAHotwiredClassBrows
WAHotwiredClassBrows
WAMethodEditorCompo
WATurboChatExample
WATurboCounter !
WATurboMultiCounter
WATurboTodo !
WATurboTodoItem !
WATurboTodoItemEdito
Filter...

instance side
actions
hooks
initialization
private
rendering
overrides

children
createTotalsPresenter
initialize
renderContentOn:
renderTotalAssignmentFormO
setCountersTo:
states
updateRoot:

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Im

WATurboM | ! Comment | initialize | + Inst. side m

```
initialize

  super initialize.
  counters := (1 to: 5) collect: [ :each |
    WATurboCounter new addDecoration: WATurboFrame new; yourself ].
  totals := self createTotalsPresenter
```

Decorate each WATurboCounter with a WATurboFrame

instantiations

TurboStream updates

Method: WATurboMultiCounter>>renderTotalAssignmentFormOn:

- Seaside-HotwireTurbo-Examp
- Seaside-InternetExplorer
- Seaside-JSON-Core
- Seaside-Pharo-Canvas
- Seaside-Pharo-Continuation
- Seaside-Pharo-Core
- Seaside-Pharo-Development
- Seaside-Pharo-Email

Filter...

- WAHotwiredClassBrowserFilel
- WAHotwiredClassBrowserNav
- WAMethodEditorComponent
- WATurboChatExample !
- WATurboCounter !
- WATurboMultiCounter !
- WATurboTodo !
- WATurboTodoItem !

Filter...

- instance side
- actions
- hooks
- initialization
- private
- rendering
- overrides

- children
- createTotalsPresenter
- initialize
- renderContentOn:
- renderTotalAssignmentFormOn:
- setCountersTo:
- states
- updateRoot:

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implementors | Sende

Dependenci | WATurboM | ! Comment | renderTota | + Inst. side m

```
renderTotalAssignmentFormOn: html

  | theValue |
  html form
    turboStreamCallback: [ :ts |
      ts replace: 'id-total' with: totals.
      counters do: [ :c | ts replaceComponent: c with: c ] ];
    with: [
      html textInput
        callback: [ :value | theValue := value ].
      html submitButton
        callback: [ self setCountersTo: theValue ] ]
```

Declare replace/update/remove/append/… actions

instantiations

Morphing

Method: WATurboMultiCounter>>updateRoot:

Seaside-HotwireTurbo-Exa
Seaside-InternetExplorer
Seaside-JSON-Core
Seaside-Pharo-Canvas
Seaside-Pharo-Continuatio
Seaside-Pharo-Core
Seaside-Pharo-Developme
Filter...

WAHotwiredClassBrowserF
WAHotwiredClassBrowserN
WAMethodEditorCompone
WATurboChatExample !
WATurboCounter !
WATurboMultiCounter !
WATurboTodo !
Filter...

instance side
actions
hooks
initialization
private
rendering
overrides

children
createTotalsPresenter
initialize
renderContentOn:
renderTotalAssignmentFormO
setCountersTo:
states
updateRoot:

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implementors

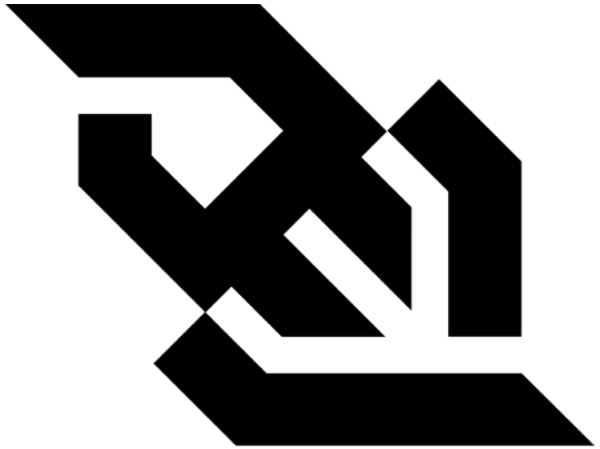
WATurboM | ! Comment | updateRoo | + Inst. side m

```
updateRoot: aRoot

  super updateRoot: aRoot.
  aRoot meta turboRefreshMethod: 'morph'.
  aRoot meta turboRefreshScroll: 'preserve'
```

Drop the frames and optimization code…
keep the single-page behavior

instantiations



Websockets

instantiations

Websockets in Seaside

http://localhost:8080/examples/websockets/counter

http://localhost:8080/examples/websockets/slider

http://localhost:8080/examples/websockets/basic-chat

Websocket Counter Example

Method: WAWebSocketCounterExample>>renderContentOn:

Seaside-WebSocket-Exam
Seaside-WebSocket-Zinc
Seaside-Welcome
Filter...

WACounterModel !
WAWebSocketBasicChatExample !
WAWebSocketCounterExample !
Filter...

instance side
accessing
actions
hooks

pusher
renderContentOn:
style
update:

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implementor

Depender | WAWebS | ! Commen | *renderC | + Inst. side

```
renderContentOn: html
  html document
    addLoadScript: (html websocket pusher: self pusher).

  html heading
    id: #count;
    with: self model count.

  html paragraph: [
    html anchor
      onClick: (html jQuery ajax callback: [ self decrease ]);
      with: 'dec'.
    html space; space; space.
    html anchor
      onClick: (html jQuery ajax callback: [ self increase ]);
      with: 'inc' ].
```

Paint Websocket connection on the page

Just an ajax callback (no re-render!)

instantiations

Websocket Counter Example

Method: WAWebSocketCounterExample class>>pusher

Seaside-WebSocket-Core
Seaside-WebSocket-Examples
Seaside-WebSocket-Zinc
Seaside-Welcome
Seaside-Widgets
Seaside-Zinc-Core
Filter...

WACounterModel !
WAWebSocketBasicChatExample
WAWebSocketCounterExample !
WAWebSocketExampleBroadcaste
WAWebSocketSliderExample !
WAWebSocketTurboExample !
Filter...

class side
accessing
class initialization
overrides

initialize
model
pusher

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implementors | Senders

Depender | Commen | WAWebS | pusher | Class side

```
pusher
  ^ Pusher ifNil: [ Pusher := WAWebSocketJSPusher new ]
```

Websocket “Javascript” pusher shared between all application sessions

Websocket Counter Example

Method: WAWebSocketCounterExample>>update:

Seaside-WebSocket-Core
Seaside-WebSocket-Examples
Seaside-WebSocket-Zinc
Filter...

WACounterModel !
WAWebSocketBasicChatExampl
WAWebSocketCounterExample
Filter...

instance side
accessing
actions
hooks

update:

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implementors | Senders

Depender | WAWebS | ! Commen | update: | + Inst. side

```
update: anAspect
  "Update the counter in all connected web browsers.
  Only the first 3 lines are really needed, the rest adds some eye-candy and can simply be removed if not
needed."

  self pusher javascript: [ :script |
    script << (script jQuery: #count)
      text: self model count;
      append: [ :html |
        html span: [
          html html: (anAspect = #increase
            ifTrue: [ '&uarr;' ]
            ifFalse: [ '&darr;' ]) ] ];
      find: 'span'; fadeOut: 'slow' ]
```

Send script to each connected client…

… that updates the value of the counter

… produces eye-candy fading of an arrow

instantiations

Websockets in Seaside

- **WAWebSocketPusher**
  - Handle websockets connections on the current (application) handler
  - Installs a WAWebSocketRequestFilter on your application url
  - E.g. Counter demo
- **WAWebSocketRequestHandler**
  - Handle websockets in a separate application handler
  - Dedicated websocket url
  - E.g. chat demo

Websocket request handler

Method: WAWebSocketExampleBroadcaster>>handleWebSocketSetup:in:

Seaside-WebComponents-
Seaside-WebComponents-
Seaside-WebDeveloper
Seaside-WebShopExample
Filter...

WARequestHandler
WAWebSocketRequestHandler
WAWebSocketExampleBroadcaster
Filter...

instan
handli
initiali
private
overri

addSocket:
handleWebSocketSetup:in:
initialize
removeSocket:
socketsDo:

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implementor

Dependen | WAWebS | Commen | *handleW | Inst. side

```
handleWebSocketSetup: aWAWebSocket in: aRequestContext
  Transcript cr; show: 'New ', aWAWebSocket printString.
  aWAWebSocket onMessage:[ :data |
    self socketsDo: [ :socket | socket send: data ] ].
  aWAWebSocket onClose:[
    Transcript cr; show: 'Closing ', aWAWebSocket printString.
    self removeSocket: aWAWebSocket ].
  aWAWebSocket onError:[ :exception |
    Transcript cr; show: aWAWebSocket.
    Transcript cr; show: exception description.
    Transcript cr; show: exception signalerContext longStack ].
  self addSocket: aWAWebSocket.
```

instantiations

Live updates: Turbostreams via Websockets

Method: WAHotwireClassBrowserSession>>initialize

Seaside-HotwireTurbo-
Seaside-InternetExplore
Seaside-JSON-Core
Seaside-Pharo-Canvas
Seaside-Pharo-Continu
Seaside-Pharo-Core
Seaside-Pharo-Develop
Seaside-Pharo-Email
Seaside-Pharo-Environ
Filter...

ManifestSeasideHotwir
WAHotwireClassBrowse
WAHotwiredClassBrows
WAHotwiredClassBrows
WAHotwiredClassBrows
WAMethodEditorCompo
WATurboChatExample
WATurboCounter !
WATurboMultiCounter
Filter...

instance side
accessing
event handling
initialization
private
overrides

handleClassChange:
handleMethodChange:
initialize
pusher
subscribeToSystemAnnounce
unregistered
unsubscribeToSystemAnoun

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs.

WAHotwire | ! Comment | initialize | + Inst. side m

```
initialize

  super initialize.
  self subscribeToSystemAnnouncements.
  pusher := WAWebSocketTurboStreamPusherOutsideOfRequest new
```

http://localhost:8080/examples/turbo/classbrowser

instantiations

Live updates: Turbostreams via Websockets

Method: WAHotwiredClassBrowserExample>>pushClassChange:to:

Seaside-HotwireTurbo-Ex
Seaside-InternetExplorer
Seaside-JSON-Core
Seaside-Pharo-Canvas
Seaside-Pharo-Continuat
Seaside-Pharo-Core
Seaside-Pharo-Developm
Filter...

ManifestSeasideHotwireT
WAHotwireClassBrowserS
WAHotwiredClassBrowse
WAHotwiredClassBrowse
WAHotwiredClassBrowse
WAMethodEditorCompon
WATurboChatExample
Filter...

instance side
accessing
event handling
hooks
initialization
rendering
overrides

initialize
listSize
pushClassChange:to:
pushMethodChange:to:
renderClassAndInstanceCont
renderContentOn:
renderPackageSearchOn:
searchTerm:
selectedClass

All Packages | Scoped View | Projects | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implen

WAHotwire | Comment | pushClassC | Inst. side m

```
pushClassChange: aClassAnnounce to: pusher

  pusher turboStream: [ :ts |
    ts
      replaceComponent: classesPane refreshList with: classesPane;
      replaceComponent: methodcategoriesPane refreshList with: methodcategoriesPane;
      replaceComponent: methodsPane refreshList with: methodsPane;
      replaceComponent: methodEditor with: methodEditor ]
```

instantiations

Release Roadmap

instantiations

Release Roadmap

- Part of Seaside 3.6.0 soon (but already on master branch)
  - Finalize implementation details
  - More complete examples and documentation
- Add Websockets for VAST Sst Adaptor
  - Planned for VAST 2026
- Future work beyond Seaside 3.6.0
  - Derive Turbostream actions from Seaside state changes (see Phoenix Liveview)
  - Seaside state management for Turbo Frames (Prune removed Seaside callbacks)
  - Zinc Websockets in GemStone

instantiations

Seaside + Hotwire + Websocket

Thank you for listening!

Questions?

Johan Brichau
VAST Consultant and Senior Software Engineer

jbrichau@instantiations.com

Contact

General Inquiry
info@instantiations.com

Sales
sales@instantiations.com

North America, Toll Free
855 476 2558

International
+1 503 263 0058