



# Advanced

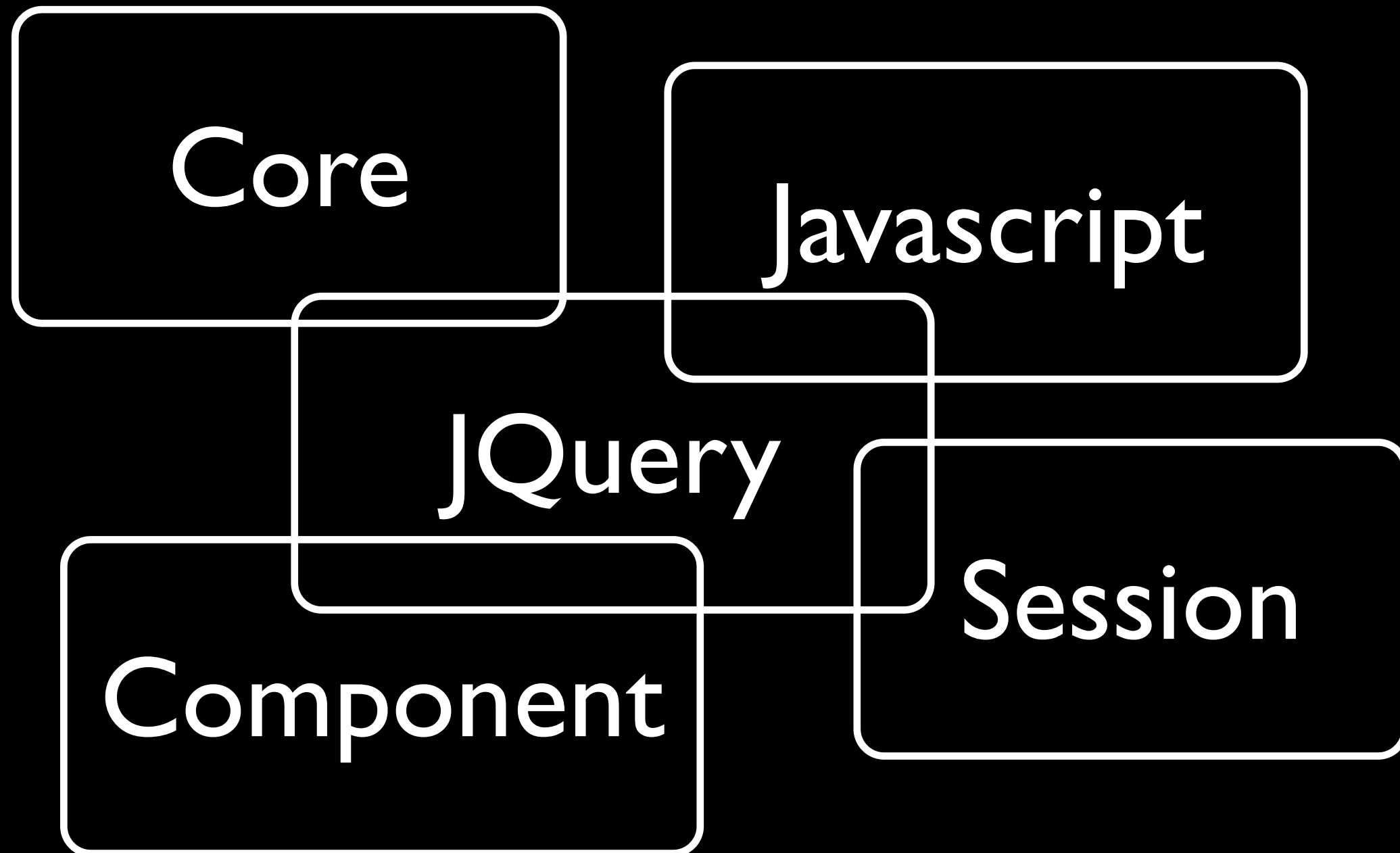
ESUG 2012 Ghent  
Aug 30, 2012

# Seaside 2.x

The word "Seaside" is centered within a white rounded rectangle with a thin white border.

Seaside

# Seaside 3.x



# What is Seaside-Core

- “Seaside Meta”
- Web Framework Framework
- HTTP Abstraction
- Servlet API
- Rack
- WSGI

# Why Seaside-Core

- portable (dialect and server)
- battle tested
- objects, not strings
- familiar objects
- negligible overhead

# Seaside 3.x

Core

Component

# Seaside 3.x

Core

Component

REST

# Seaside 3.x

Core

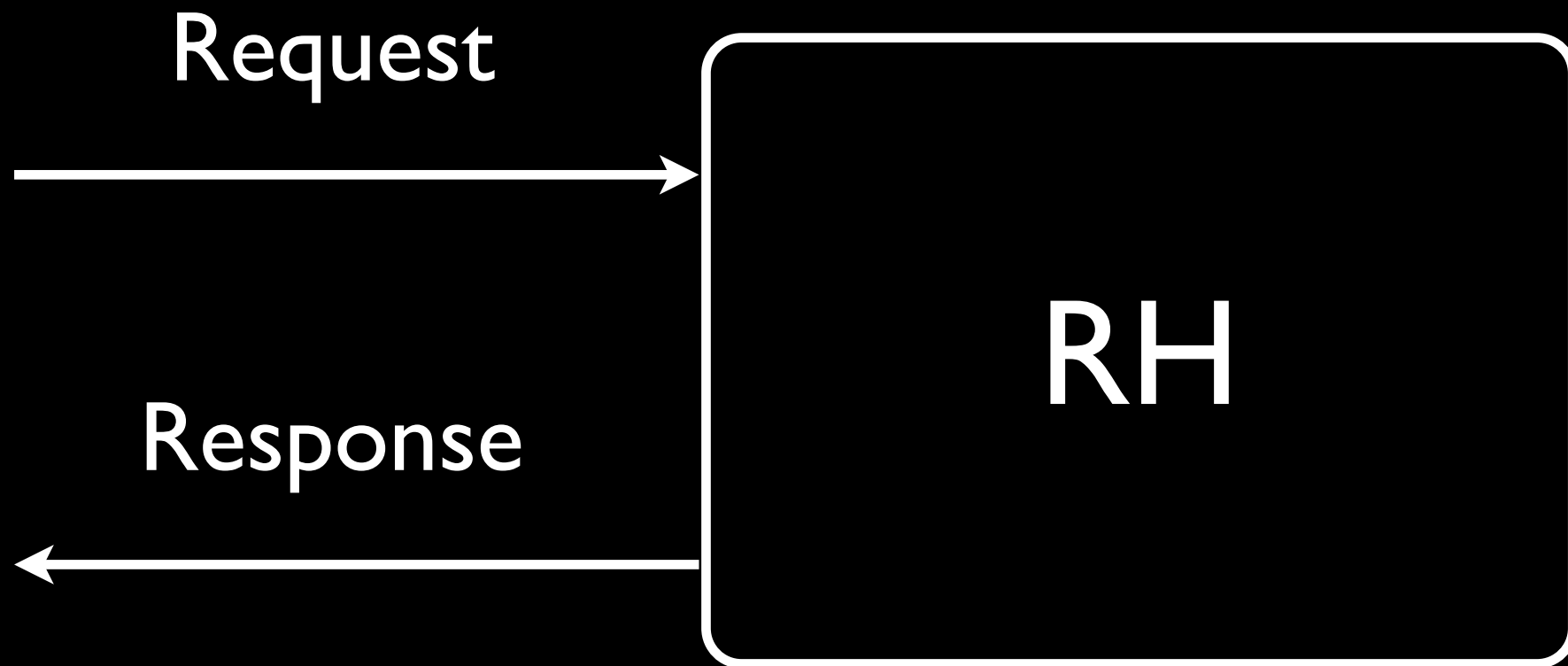
Component

REST

...



# WARequestHandler

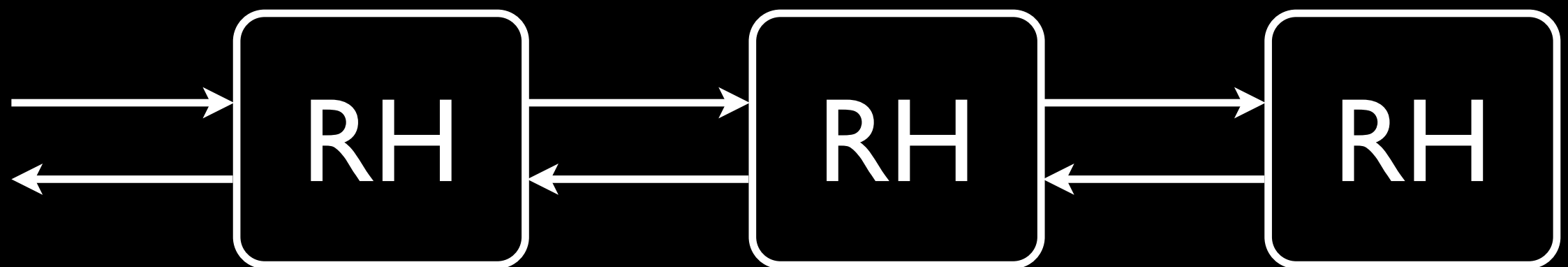


# WARequestHandler

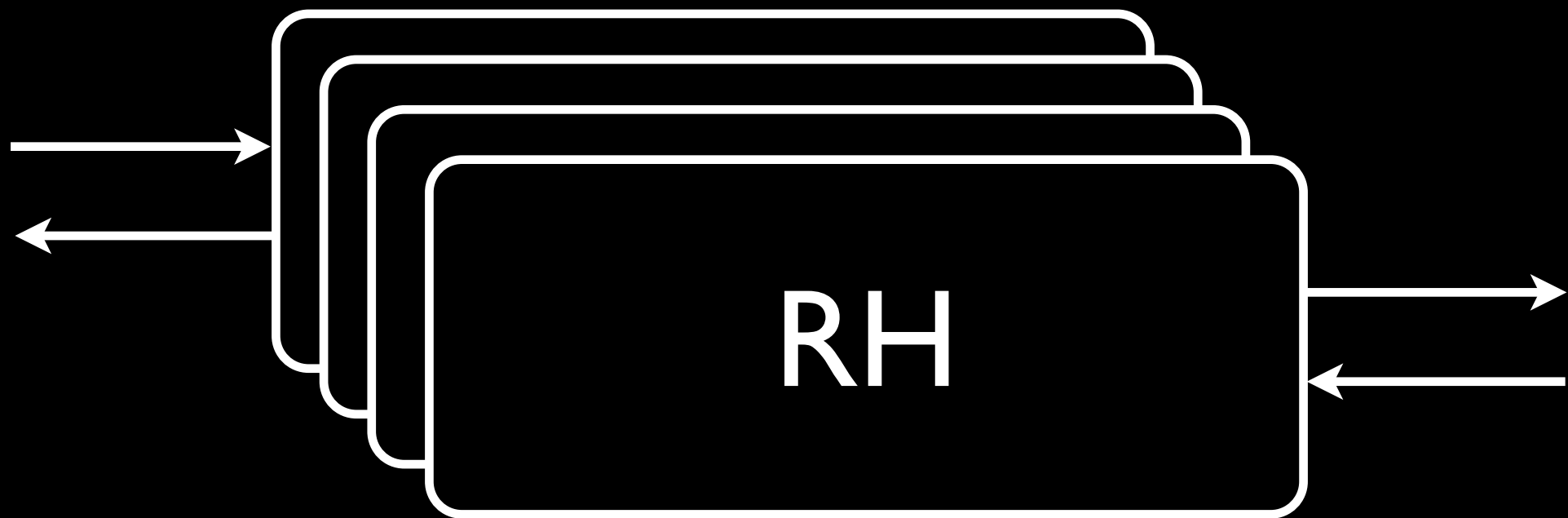
/conferences

/esug

/2012



# WARequestFilter



# WASession 2.8

- current request
- jump to
- last continuation
- continuations
- monitor
- ...

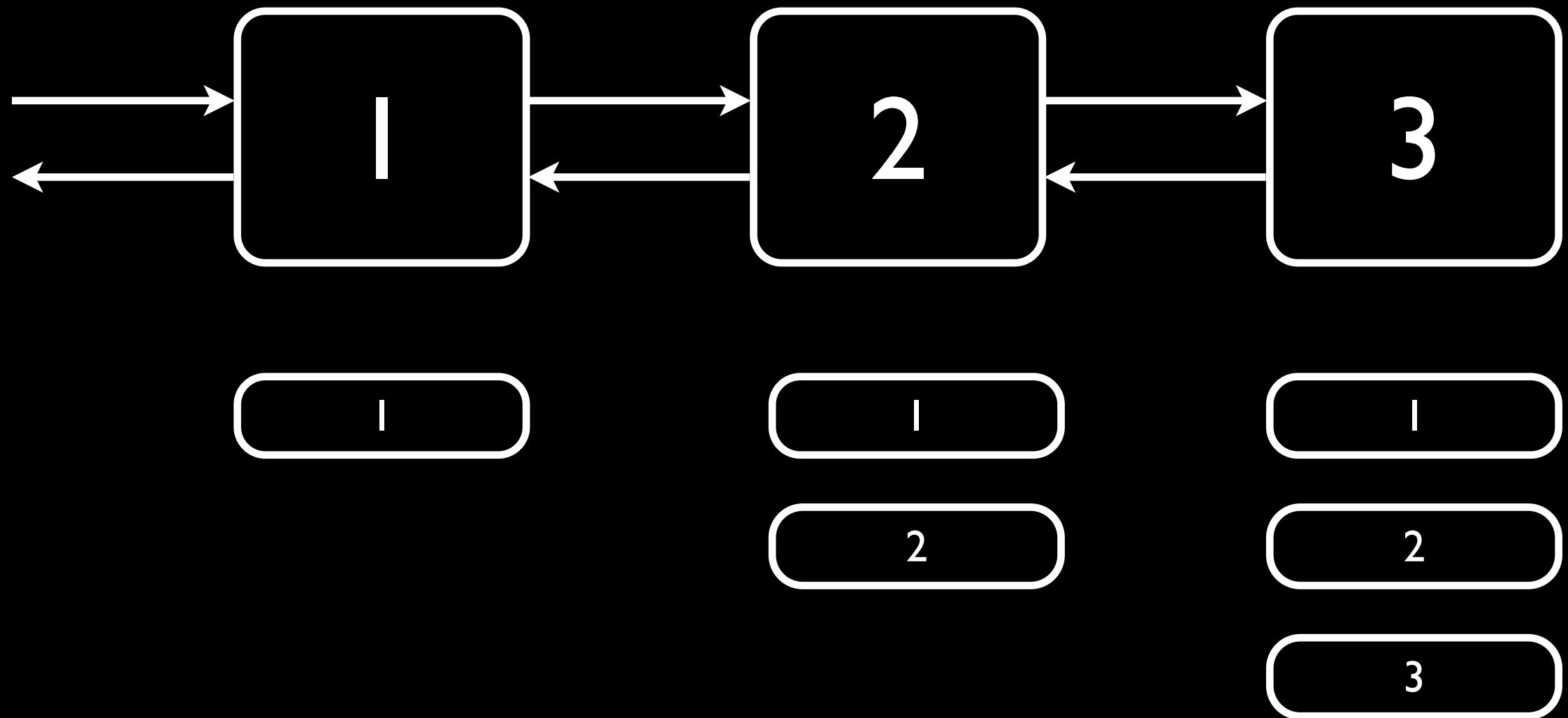
# WASession 3.x

- continuations
- properties
- (document handlers)

# WARequestContext

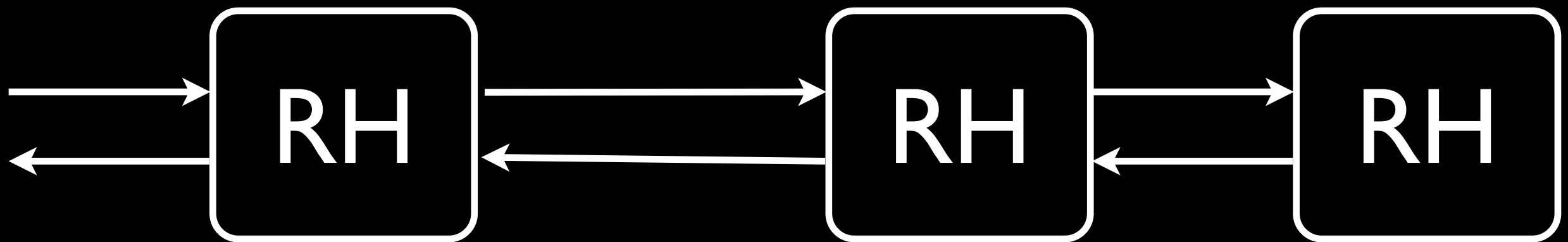
- request
- response
- handler stack

# Request Handler Stack



# WAPathConsumer

/conferences/esug/2012/...



/esug/2012/...

/2012/...

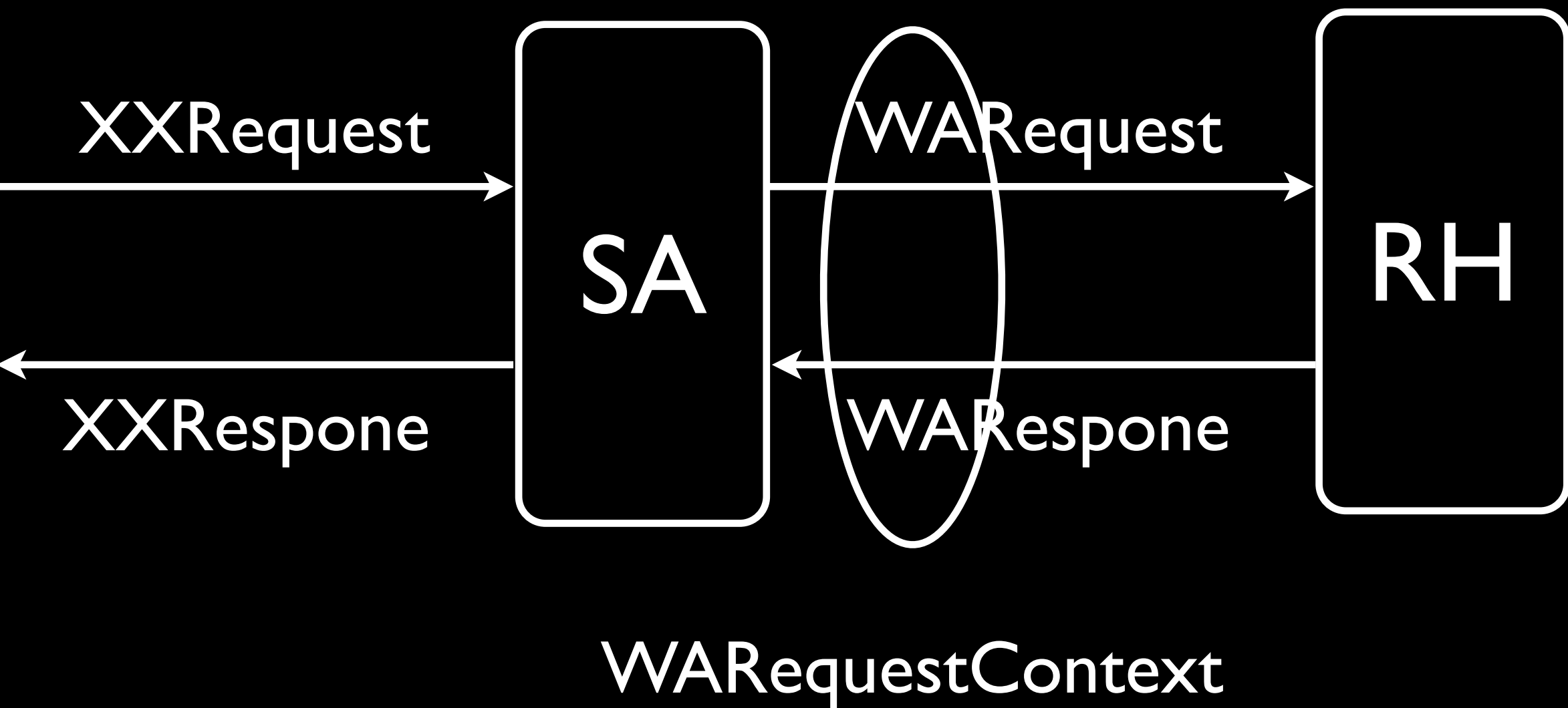
/...



# WADispatcher

‘conferences’ -> aRequestHandler  
    ‘sponsors’ -> aRequestHandler  
        ‘...’ -> aRequestHandler

# WAServerAdapter



# Frank

a micro framework inspired by Sinatra

# Seaside-REST

# Disclaimer

- REST in this context simply means “pretty URLs”
- pretty URLs for web services, not web applications
- this will feel a little weird

index

<get>

^ 'Hello World'

index

<get>

<path: 'index.html'>

^ 'Hello World'

indexHtml

<get>

<path: 'index'>

<produces: 'text/html'>

^ '<h1>Hello World</h1>'

Accept: text/html



indexXml

<get>

<path: 'index'>

<produces: 'text/xml'>

^ '<hello><world/></hello>'

Accept: text/xml

uploadXml

<post>

<consumes: 'text/xml'>

...

Content-Type: text/xml

uploadCsv

<post>

<consumes: 'text/csv'>

...

Content-Type: text/csv

getAllStartingAt: start limit: limit named: name

<get>

<path: ‘/{name}/\_all\_docs?start={start}&limit={limit}’>

...

# #shouLdCacheRoutes

# WARestfulComponentFilter

```
count: count
```

```
<get>
```

```
<path: '/{count}'>
```

```
self startSessionWithRoot: (WACounter new  
    count: count greaseInteger;  
    yourself)
```

# Data Binding

- XML or JSON canvas
- Magritte-XML
- Magritte-JSON

# Dialects

- GemStone
- Pharo
- VAST



# Seaside 3.1

# Session Tracking Fully Customizable

- query fields
- cookie only
- cookie if supported, query field otherwise
- cookie for browser, IP for crawler
- SSL session id (\*)
- path parameter (\*)

# Path Parameter

- `/;_s=KAAWI0x3c6KLnN6Q`
- doesn't need to be hidden form parameter
- required by some load balancers

# WAMain gone

- specify initial continuation
- subclass `WASessionContinuation` if you used to subclass `WAMain`

# WAPluggableActionContinuation

- replace WAActionPhaseContinuation subclasses with WACallbackProcessingActionContinuation subclasses

# JSON

- separate from JavaScript
  - various subtle bugs fixed
- canvas API

# Streaming Redone

- on demand flushing
- continuation that flushes after rendering `</head>`
- better portability to other servers

# HTML 5

● #multipleValuesCallback:



# Document Handlers

- stored in session
  - no longer stored alongside sessions
- `_d`

# Other News

- Nick now part of core team
- experimental Pharo 2.0 support
- walkback should open at correct place in VW
- platforms can implement optimized HTML escaping

# Help Wanted

- Morphic Control panel for Pharo 2.0

# Seaside Sprint

- Friday at Vooruit
- Saturday at Camp location
- (Sunday)

# Seaside Sprint Sponsors

- Reza Razavi
- 2Rivers

# Links

- <https://github.com/marschall/esug-2012-presentation>
- <http://ss3.gemstone.com/ss/frank>
- <http://code.google.com/p/seaside/wiki/Seaside310Changelog>
- <http://book.seaside.st/book/advanced/restful>
- <http://code.google.com/p/seaside/wiki/SeasideRest>