StrataCode

Do more than code - StrataCode

formerly called LayerCake

Project Overview

- Jeff Vroom: 20 years of building platforms (AVS, ATG, and Adobe's BlazeDS and Flex Data Services)
- Selected software patterns that improve software efficiency
- Last 5 yrs - StrataCode and tech due diligence
- Usable today, seeking alpha testers and partners
- Open source release soon!

Mission: Improve Software Efficiency

- Layers to organize code, data, files
- ATG built on layered component configuration
- But way more than just layers…
- Improvements for business and engineering

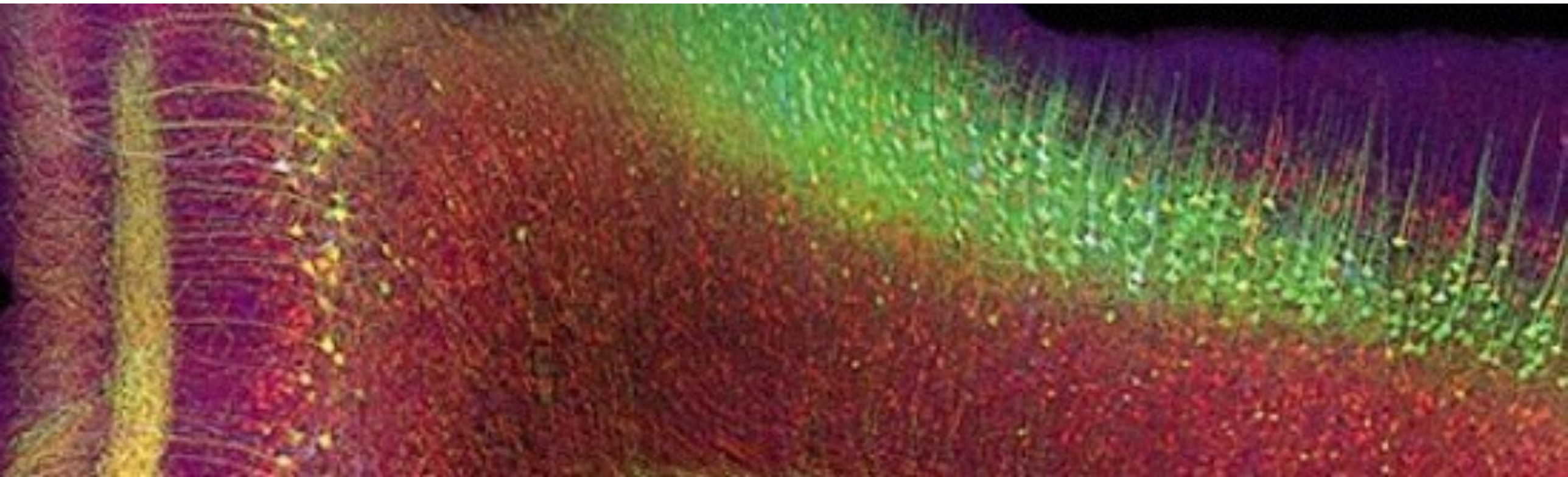
More Efficient for Business

- More people in control over features
- Faster, easier customizations
- Higher fidelity integrations:
  - Synchronized data models
  - Filtering, workflow
- Improved packaging and deployment for product feature lines

KHL Design Studio

Put Business in Control

- Customized management UIs
- Live editing of business rules (ala Excel), deployable at scale
- Workflow built into layers: code, data, files



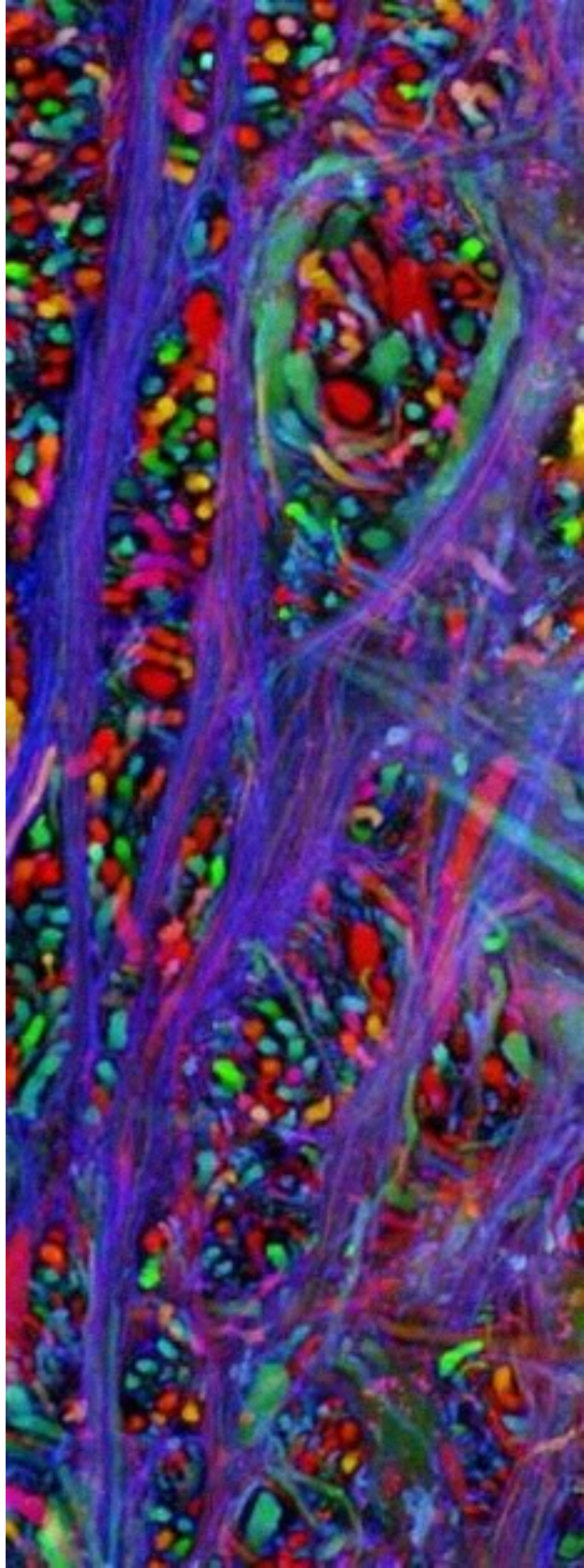
More Efficient for Engineering

- Less code copying
- More readable and manageable applications by separating framework code
- Reactive, declarative code - less does more
- Just the right patterns for readable, reliable, fast code
- Low latency, real time built in
- Scaleable - code, users, data

Youtube - 30 story building in 15 days

More Engineering Benefits

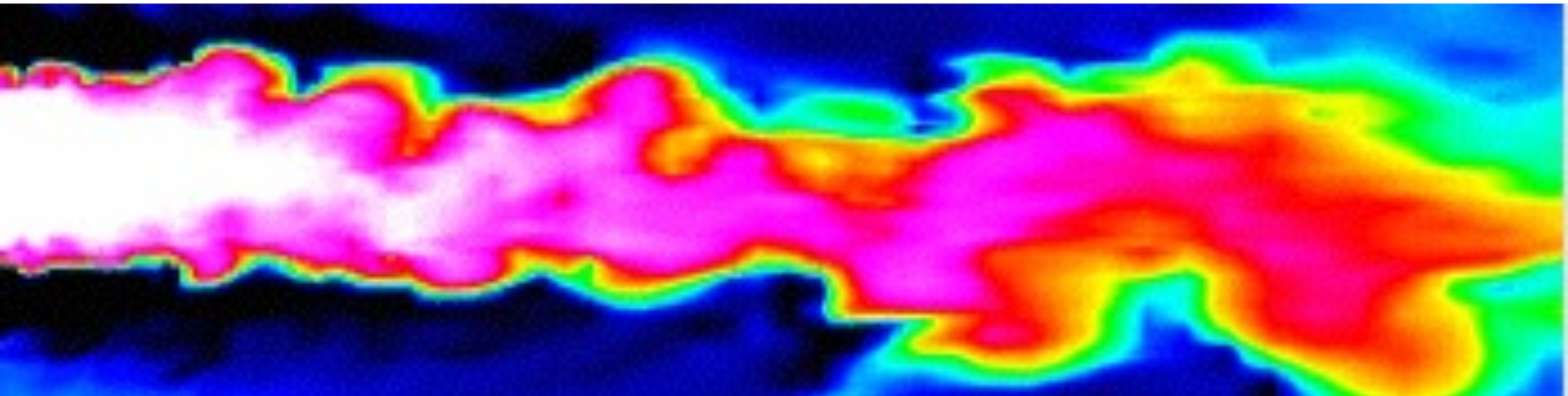
- One language: client or server
- That’s not Javascript
- More control and features from “the code skeleton”:
  - object instances with life cycles (aka scopes)
  - get/set property and references
- Robust stateful client/server applications
- Live programming with a static typed language



Brain nerve fibers, Jeff Lichtman Harvard

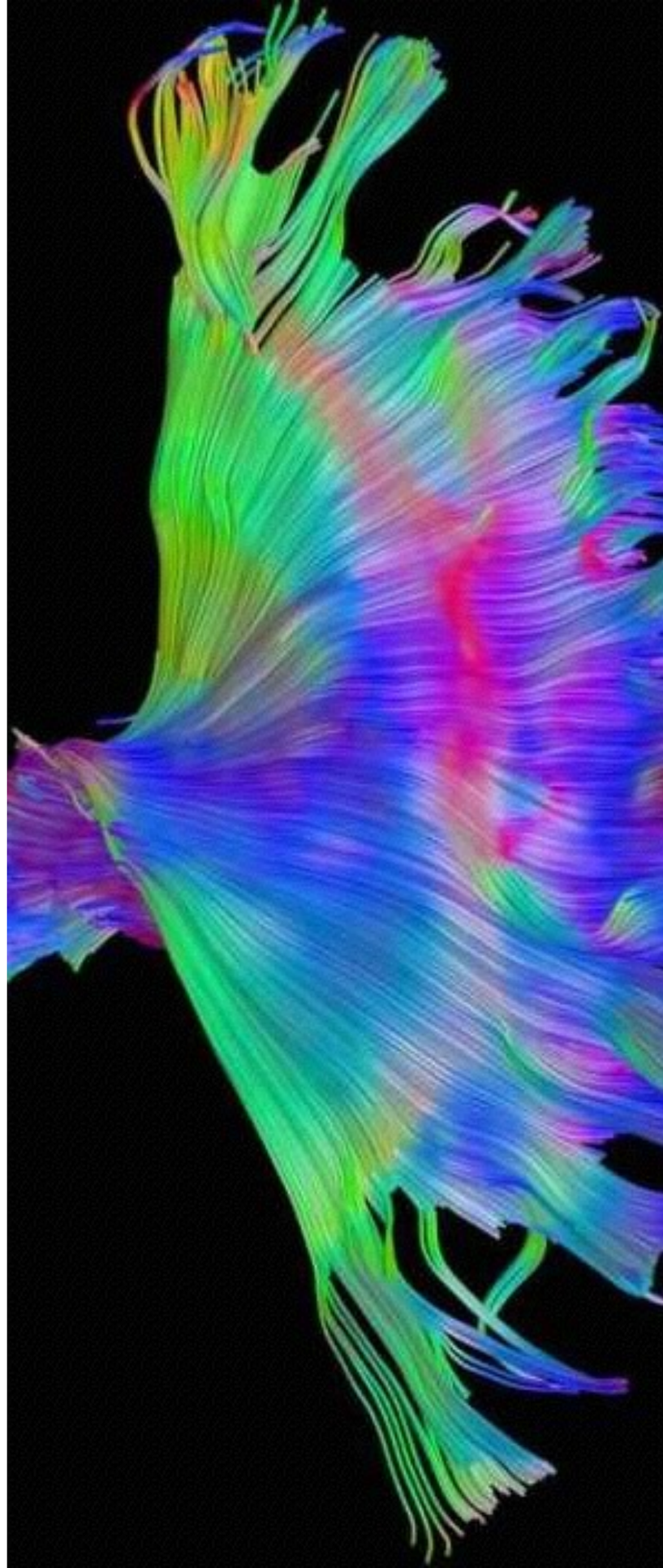
What is StrataCode?

- Java based code pre-processor
- Frameworks and integrations
- Java to Javascript, components, data binding, multiple inheritance



Layers: More Flexible Objects

- Simple organizational extension to Java
- Similar to “delta oriented programming”
- Each layer is a directory of source files
- Like names merged (modify) or replaced (class/object)
- Modify types using plain old inheritance



nerve fibers, Walt Schneider Laboratory

Uses for Layers

- Compatible refactoring without code copying
- Patch files - “monkey patching”
- Organize code by dependency for maximum reuse - browser, mobile, desktop, server, …
- Organize by function - UI, database, etc.
- Client/server synchronization: isomorphic
- Incremental, universal customization - code, data, files

mouse brain hippocampus layers, Carl Schoonover,

Data Binding

- When b changes, set a

```
a := b;
```

(b is any Java expression)

- When a changes eval or set b

```
a =: b;
```

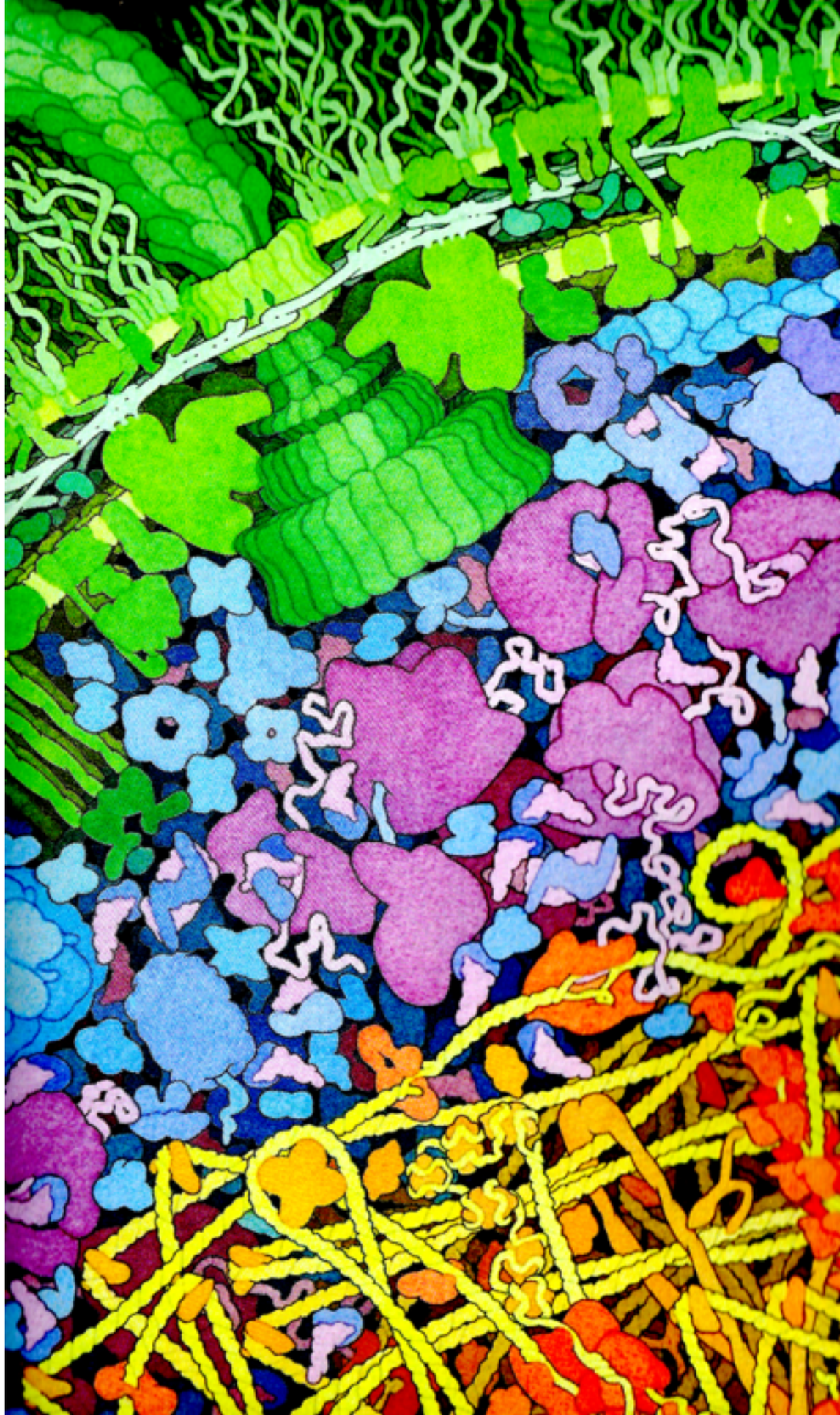
(b is almost any expression)

- When a or b changes update the other

```
a :=: b;
```

(b must be invertible, e.g.

`c * 7` is ok but `c * d` is an error)



layered structure of a cell, Machinery of Live, David Goodsell

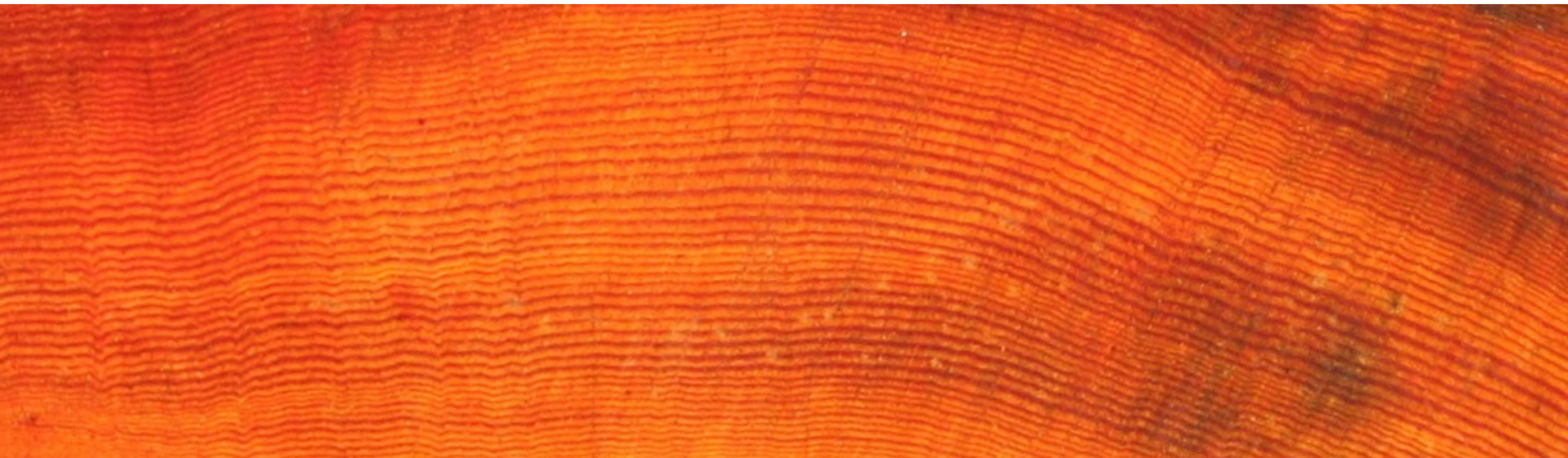
Code Processing - Frameworks and Integrations

- Parselets: state of the art in code processing
  - Parse via declarative grammar to model (AST)
  - Change model, code updates incrementally
    - Read, modify, write code from frameworks and tools
- Translate between formats (e.g. Java to JS)
- Better frameworks and integrations = less code for more

Peutinger Table - visual representation of the transportation network of the Roman Empire.

Learn More

- Watch the StrataCode Web Demo on YouTube
- Go to www.stratacode.com for more info
- Email jeff@jvroom.com



StrataCode Web Demo: Preview

Jeff Vroom

Layered code organization and management UIs

Note: LayerCake (lc) => StrataCode (sc) - renaming in progress

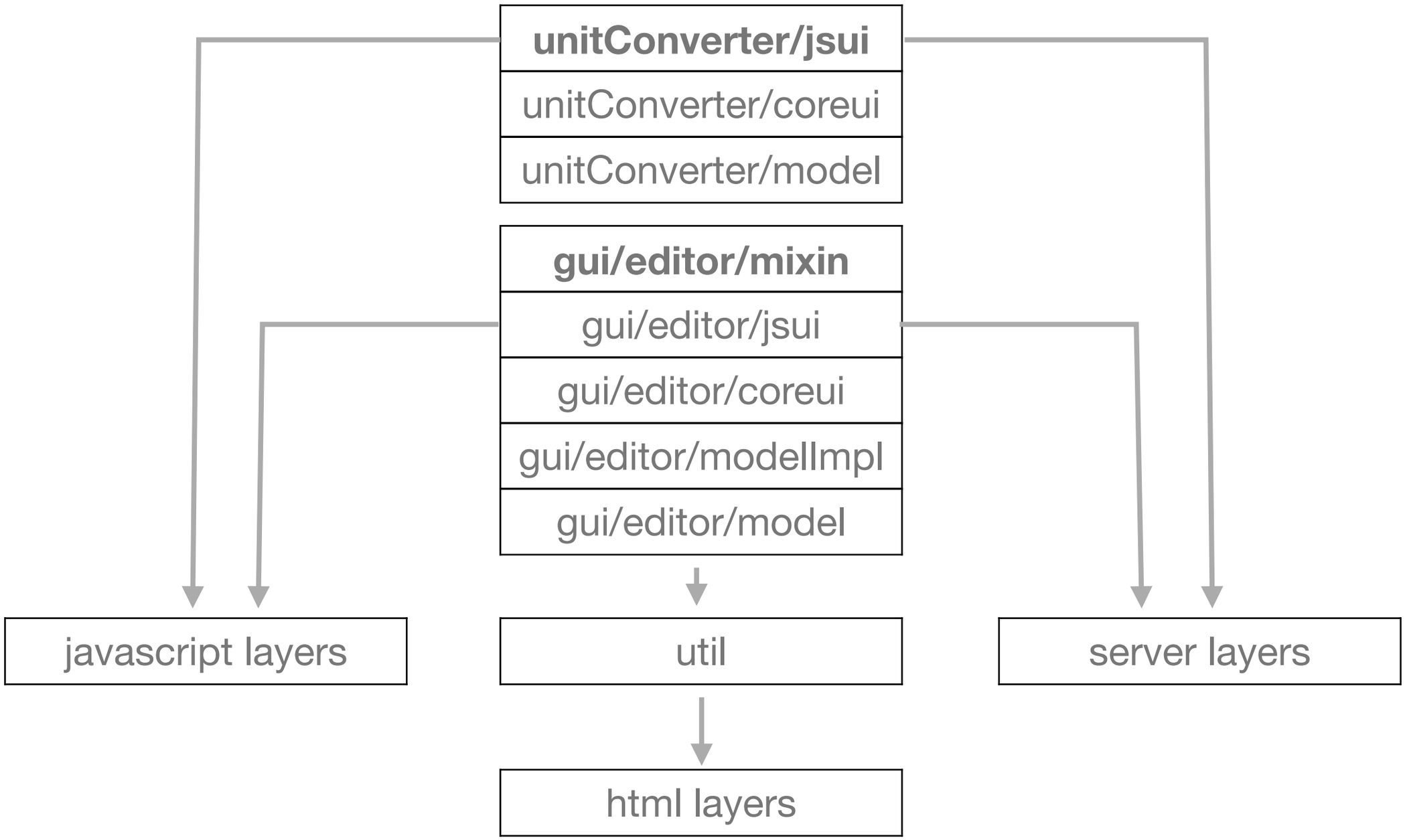
Run StrataCode with two layers:

```
% sc gui/editor/mixin unitConverter/jsui
```

unitConverter/jsui

gui/editor/mixin

All Dependent Layers Are Sorted



And Self-Organize Into Runtimes

client only

Javascript Runtime

- unitConverter/jsui
- unitConverter/coreui
- unitConverter/model
- gui/editor/mixin
- gui/editor/jsui
- gui/editor/coreui
- gui/editor/model
- javascript layers
- util
- html layers

Java

Javascript

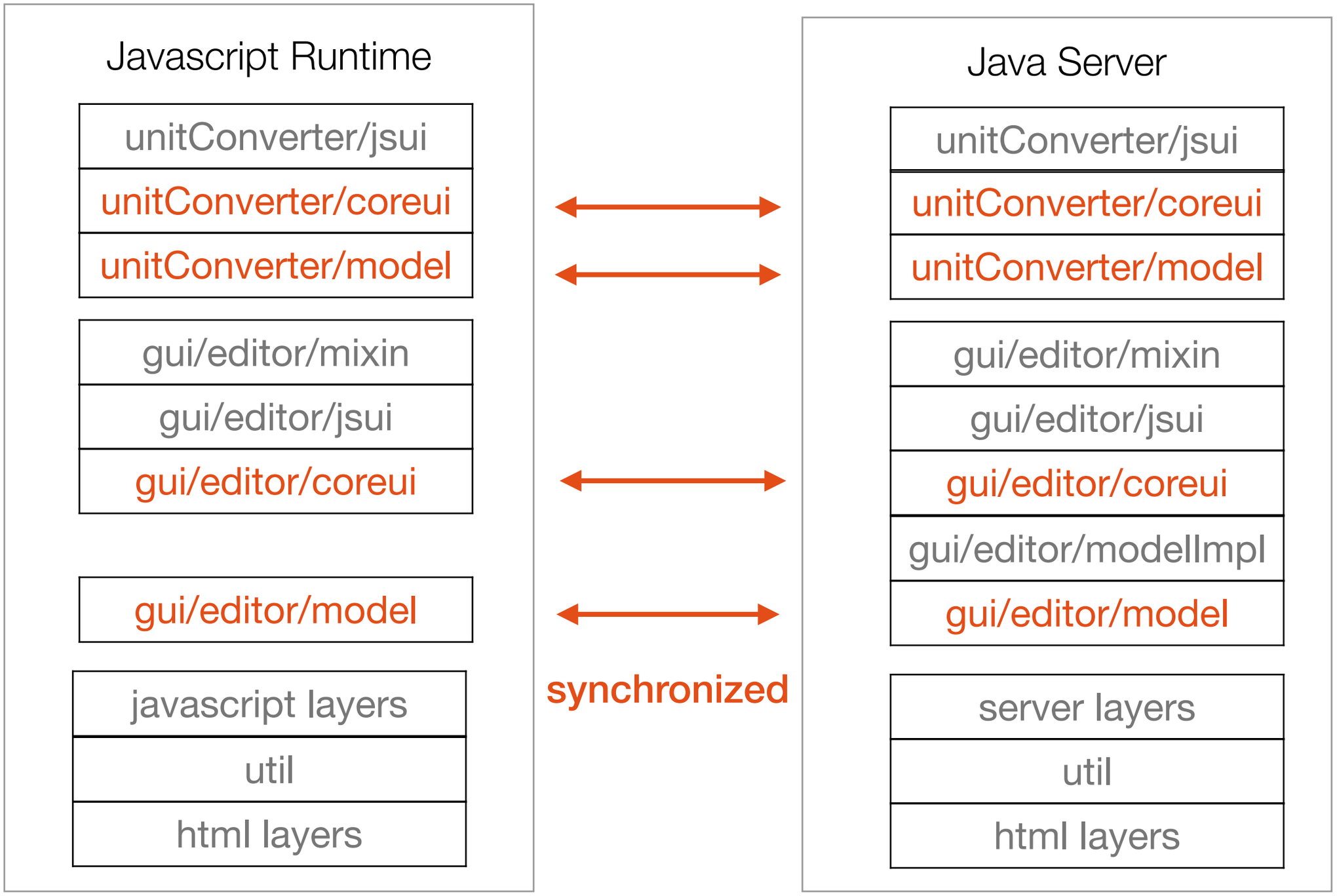
Java Server Runtime

- unitConverter/jsui
- unitConverter/coreui
- unitConverter/model
- gui/editor/mixin
- gui/editor/jsui
- gui/editor/coreui
- gui/editor/modelImpl
- gui/editor/model
- server layers
- util
- html layers

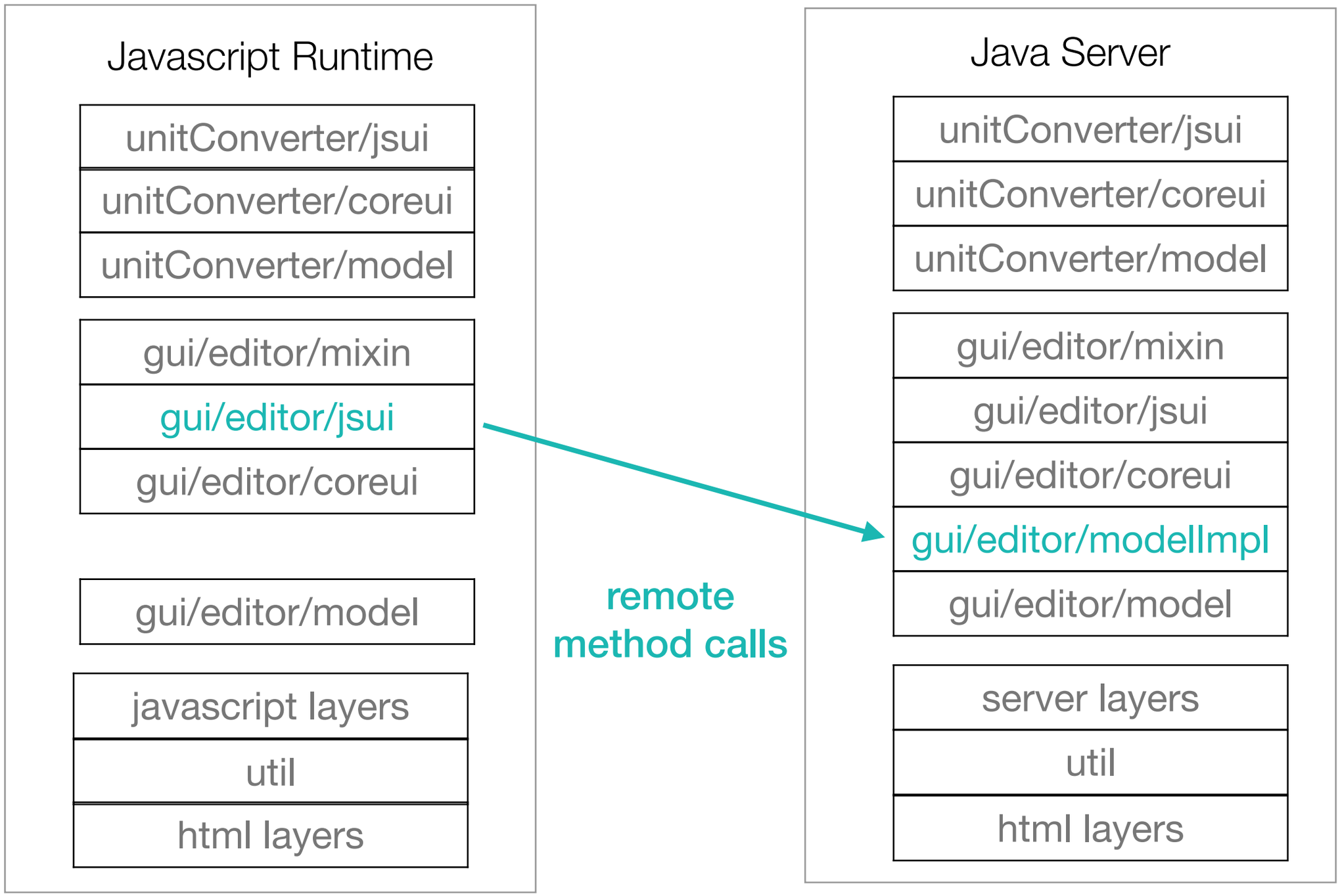
server only

Java

Overlapping Layers are Synchronized

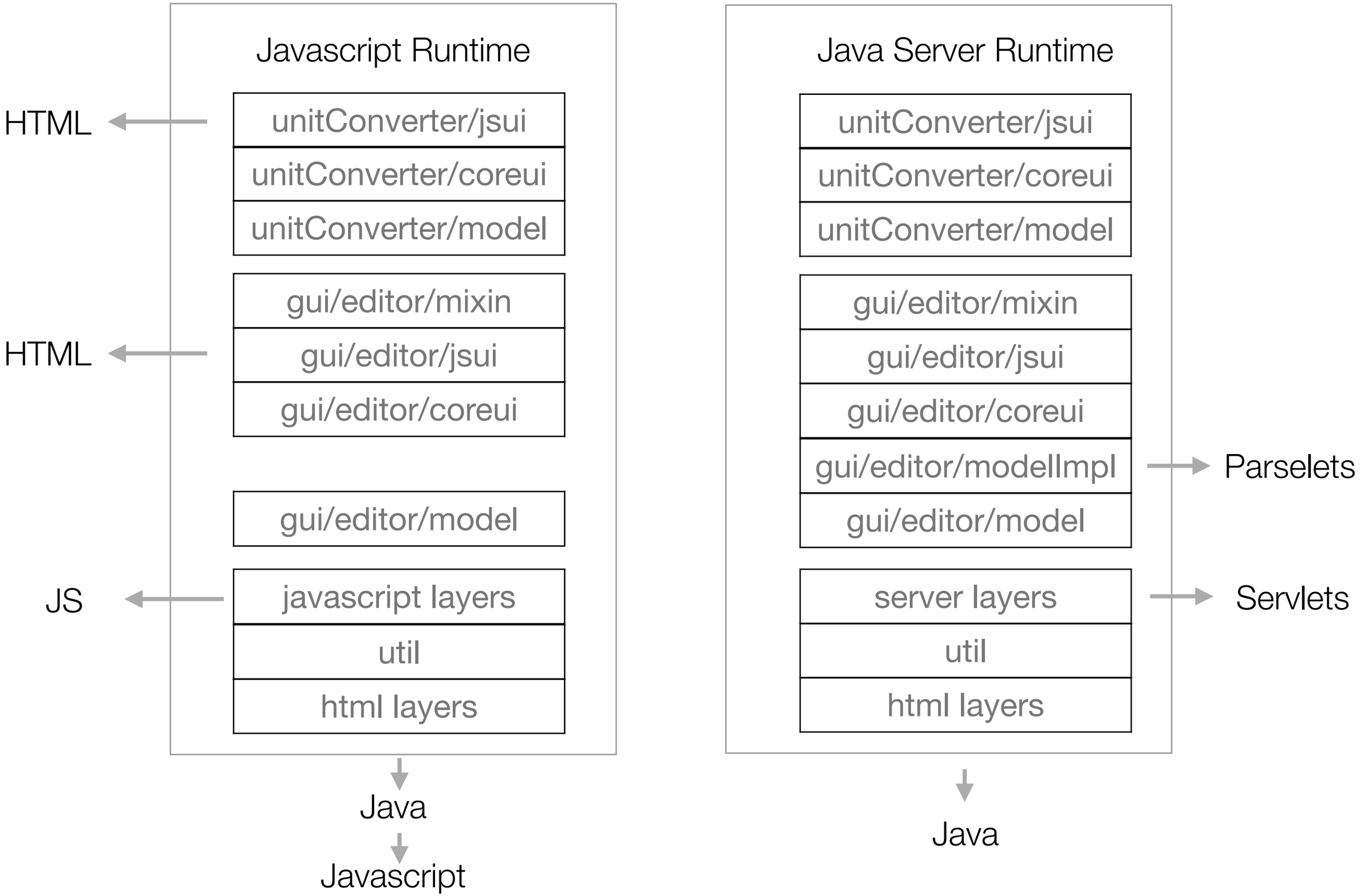


Automatic Remote Methods With Bindings



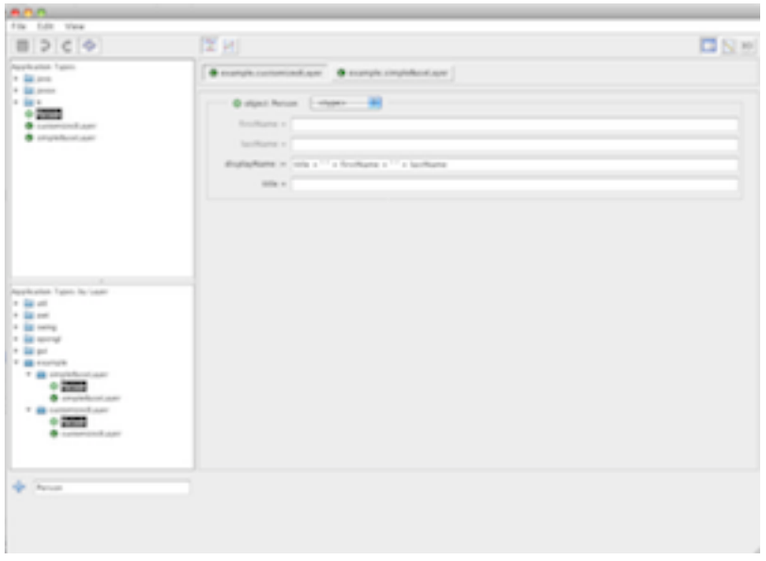

```
<span clickEvent='=: editorModel.ctx.save()'>

<form submitEvent"=: errorText = editorModel.setElementValue(type, instance, propC,
                    textField.value, updateInstances, instance == null)">
```

Code Organized By Dependencies

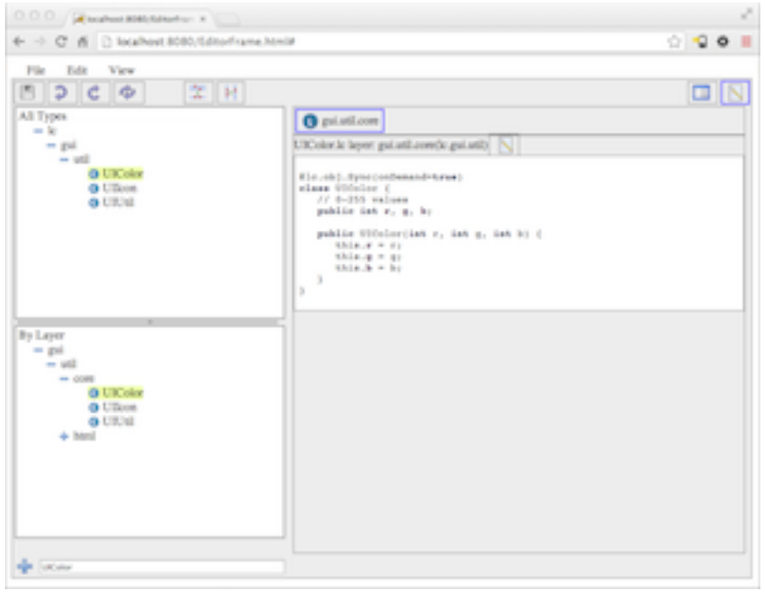


Reuse More Code

Desktop Editor



Client/Server Editor



lines of code

desktop UI				client/server UI
gui/editor/swingui	2700	1100	gui/editor/jsui	
gui/editor/coreui	430		gui/editor/coreui	
gui/editor/modelImpl	2150		gui/editor/modelImpl	
gui/editor/model	1100		gui/editor/model	
total shared	3680			
% shared	57%	76%		

Dynamic Layers:

% sc gui/editor/mixin -dyn unitConverter/jsui

gui/editor/mixin

unitConverter/jsui

Live Client/Server Programming