Configuring in the Browser, Really!

Tim Geisler, Heribert Schütz

webXcerpt Software GmbH

tg@webxcerpt.com, hs@webxcerpt.com

CWG 2015, Prague 2015-04-28

Our History

- Product configuration since 2002, with SAP since 2007
- Built and maintained
  - Models
  - Modeling environments
  - Configuration frameworks

Problem 1: Modeling

Problem 1: Modeling

- Framework-specific modeling tools

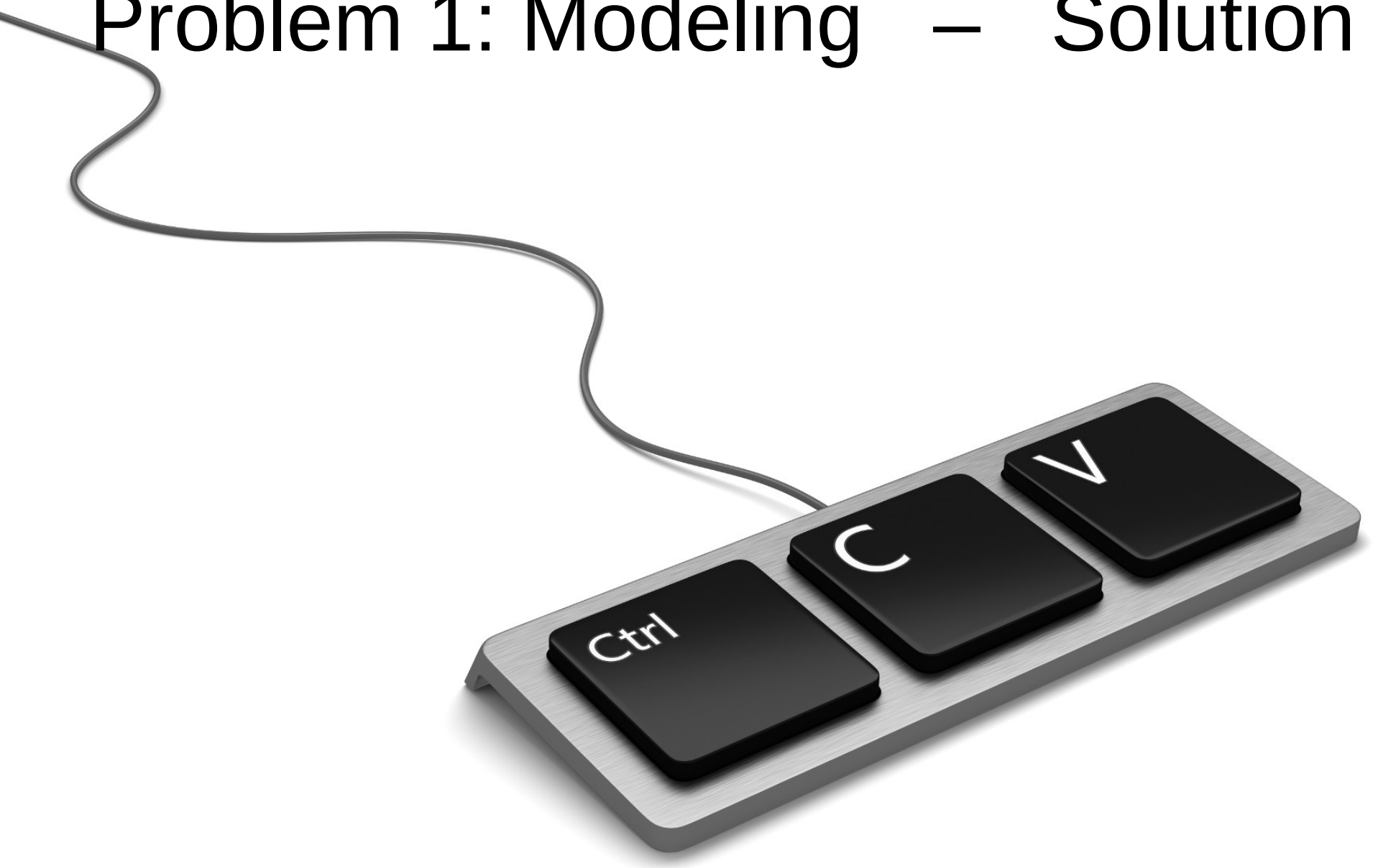
Problem 1: Modeling

- Framework-specific modeling tools
- Lack of abstraction features and data structures
  - Loops, functions
  - Arrays, objects (with methods)

Problem 1: Modeling

- Framework-specific modeling tools
- Lack of abstraction features and data structures
  - Loops, functions
  - Arrays, objects (with methods)
- Models not represented as human-readable text
  - Edit, search & replace
  - Discuss, annotate
  - Compare, manage revisions

Problem 1: Modeling – Solution A



Problem 1: Modeling – Solution B

Problem 1: Modeling – Solution B

Our solution so far:

Problem 1: Modeling – Solution B

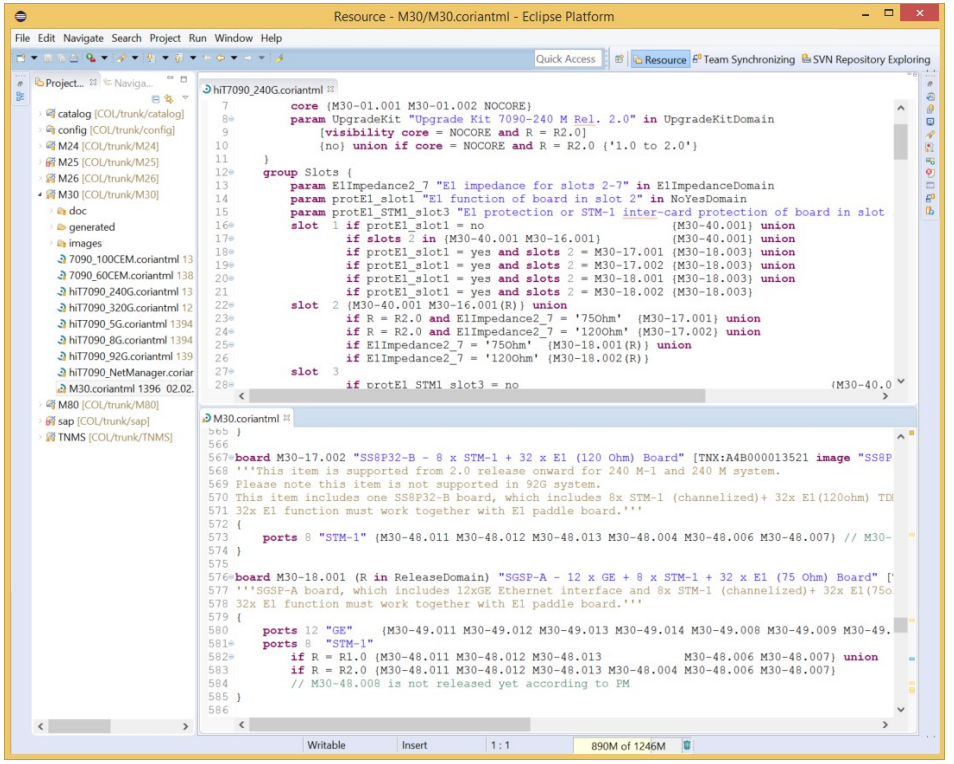
Our solution so far:

- Customer-specific modeling languages

Problem 1: Modeling – Solution B

Our solution so far:

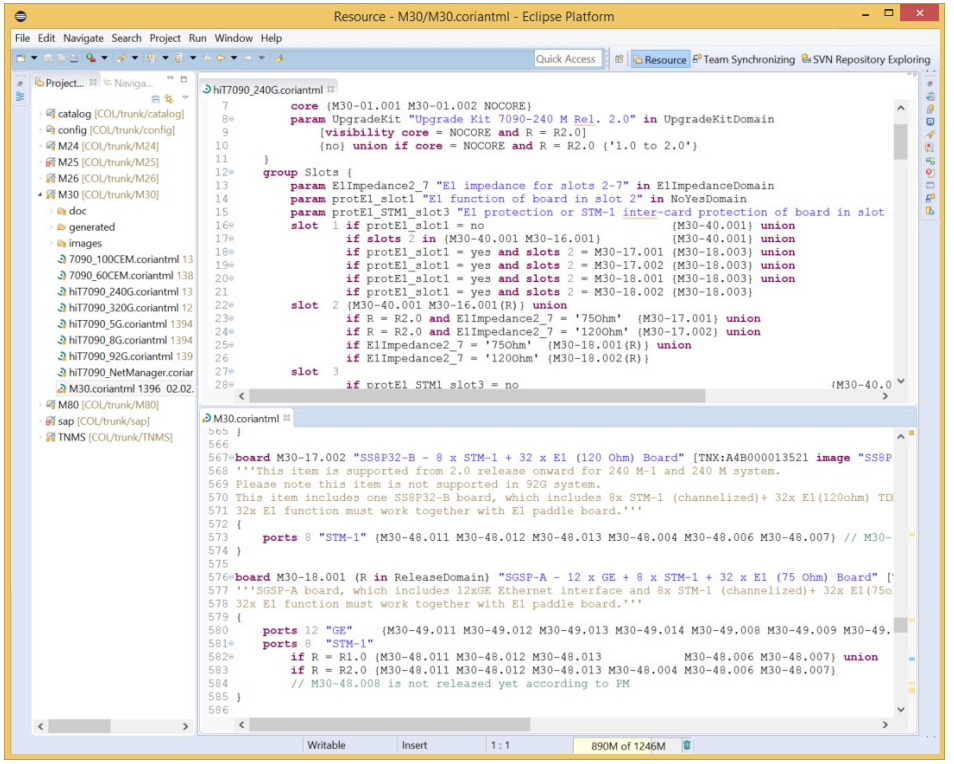
- Customer-specific modeling languages
- Modeling environments based on Eclipse and Xtext



Problem 1: Modeling – Solution B

Our solution so far:

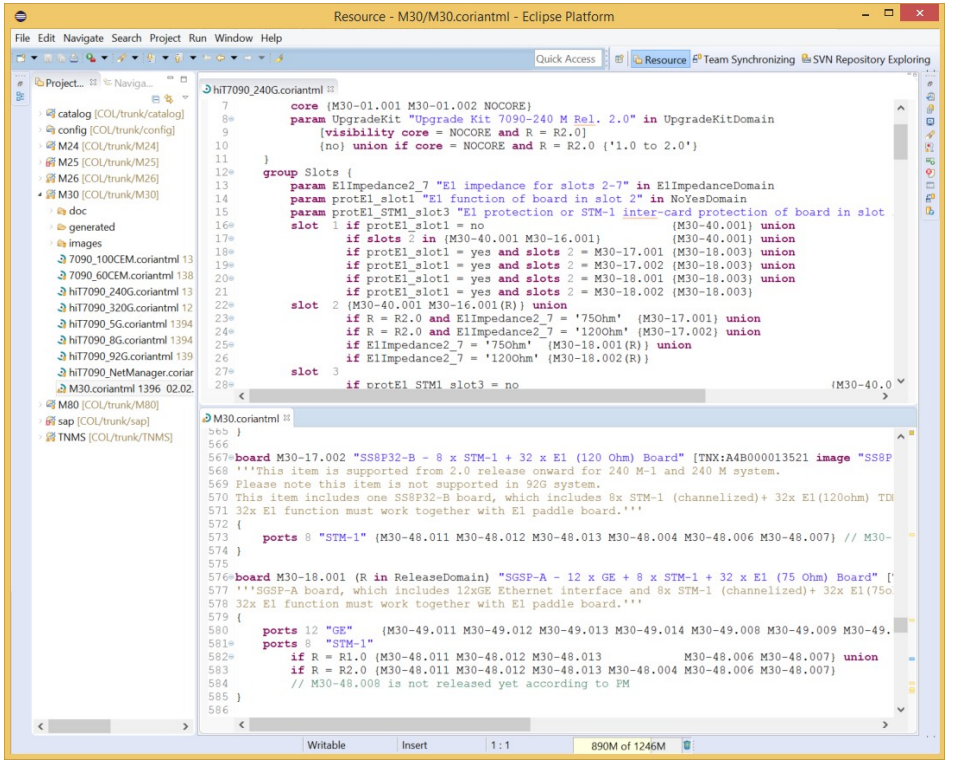
- Customer-specific modeling languages
- Modeling environments based on Eclipse and Xtext
- Automated generation of for target framework



Problem 1: Modeling – Solution B

Our solution so far:

- Customer-specific modeling languages
- Modeling environments based on Eclipse and Xtext
- Automated generation of model representation for target framework
- See also CWG talks
  - Vienna 2010:
    ConfigModeler and VClipse - languages and IDEs for product modeling
  - Cologne 2011:
    Domain-Specific Languages for Product Modeling
  - Berlin 2012:
    How to Build Your Own Product-Modeling Environment?



Problem 1: Modeling – Solution C

Problem 1: Modeling – Solution C

- Use a programming language
  - For application-specific inferencing
  - But also to build up the model

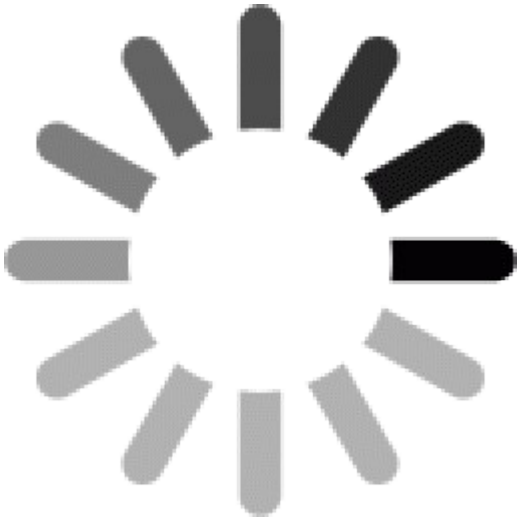
Problem 1: Modeling – Solution C

- Use a programming language
  - For application-specific inferencing
  - But also to build up the model
- Use programming tools
  - Editors/IDEs
  - Debuggers and profilers
  - Revision control
  - Test and CI frameworks

Problem 1: Modeling – Solution C

- Use a programming language
  - For application-specific inferencing
  - But also to build up the model
- Use programming tools
  - Editors/IDEs
  - Debuggers and profilers
  - Revision control
  - Test and CI frameworks
- General purpose tools and languages
  - Maturity
  - Re-usable knowledge, may already be available
  - Large communities and „ecosystems“

Problem 2: User Experience



Problem 2: User Experience

Problem 2: User Experience

- Performance

Problem 2: User Experience

- Performance
  - Client-server round trips

Problem 2: User Experience

- Performance
  - Client-server round trips
- Rigid UI
  - UI structure imposed by framework
  - High costs for application-specific UI

Problem 2: User Experience

- Performance
  - Client-server round trips
- Rigid UI
  - UI structure imposed by framework
  - High costs for application-specific UI
- Need to be online

Problem 2: User Experience

- Performance
  - Client-server round trips
- Rigid UI
  - UI structure imposed by framework
  - High costs for application-specific UI
- Need to be online

Increasing gap:

Configurators ↔ Modern web applications

In the meantime ...

In the meantime ...

In the meantime ...

Client hardware improved

In the meantime ...

Client hardware improved

- CPU
- Memory

In the meantime ...

Client hardware improved

- CPU
- Memory
- Even on mobiles

In the meantime ...

Client hardware improved

- CPU
- Memory
- Even on mobiles

… but the speed of light remained the same.

In the meantime ...

In the meantime ...

Browser improvements:

In the meantime ...

Browser improvements:

- JavaScript performance

In the meantime ...

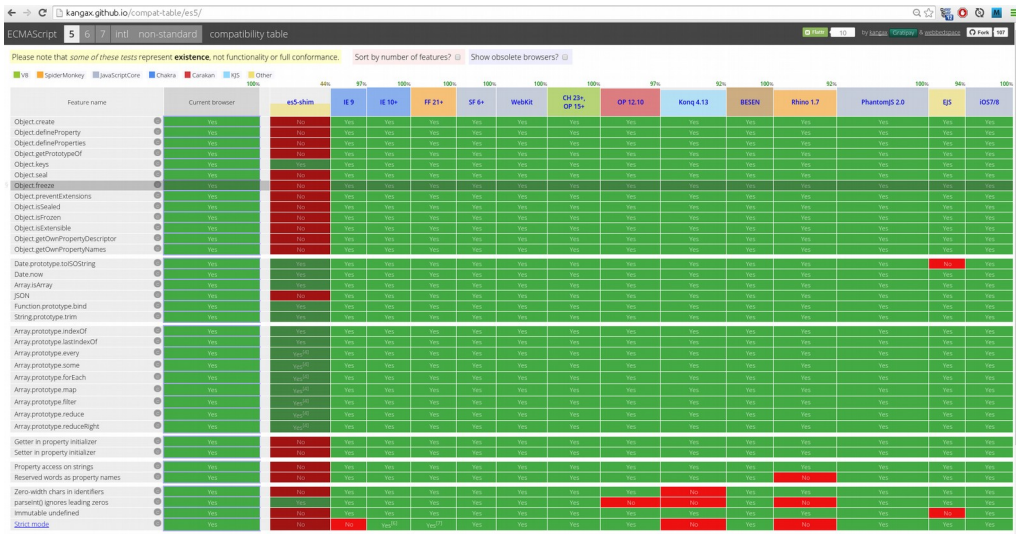
Browser improvements:

- JavaScript performance
- Standardized features 
  - UI extensions
  - Offline applications
  - Local storage
  - ...

In the meantime ...

Browser improvements:

- JavaScript performance
- Standardized features
  - UI extensions
  - Offline applications
  - Local storage
  - ...
- Improved compatibility



http://kangax.github.io/compat-table/es5/

In the meantime ...

A software ecosystem for web applications flourished:

- Web-application frameworks
- Preprocessors for JavaScript/HTML/CSS
- Libraries
- Build tools
- ...

In the meantime ...

In the meantime ...

Web browsers have become a serious application platform.

In the meantime ...

Web browsers have become a serious application platform.

Even for the business logic.

In the meantime ...

Web browsers have become a serious application platform.

Even for the business logic.

And they are getting better and better.

Configuring in the Browser:

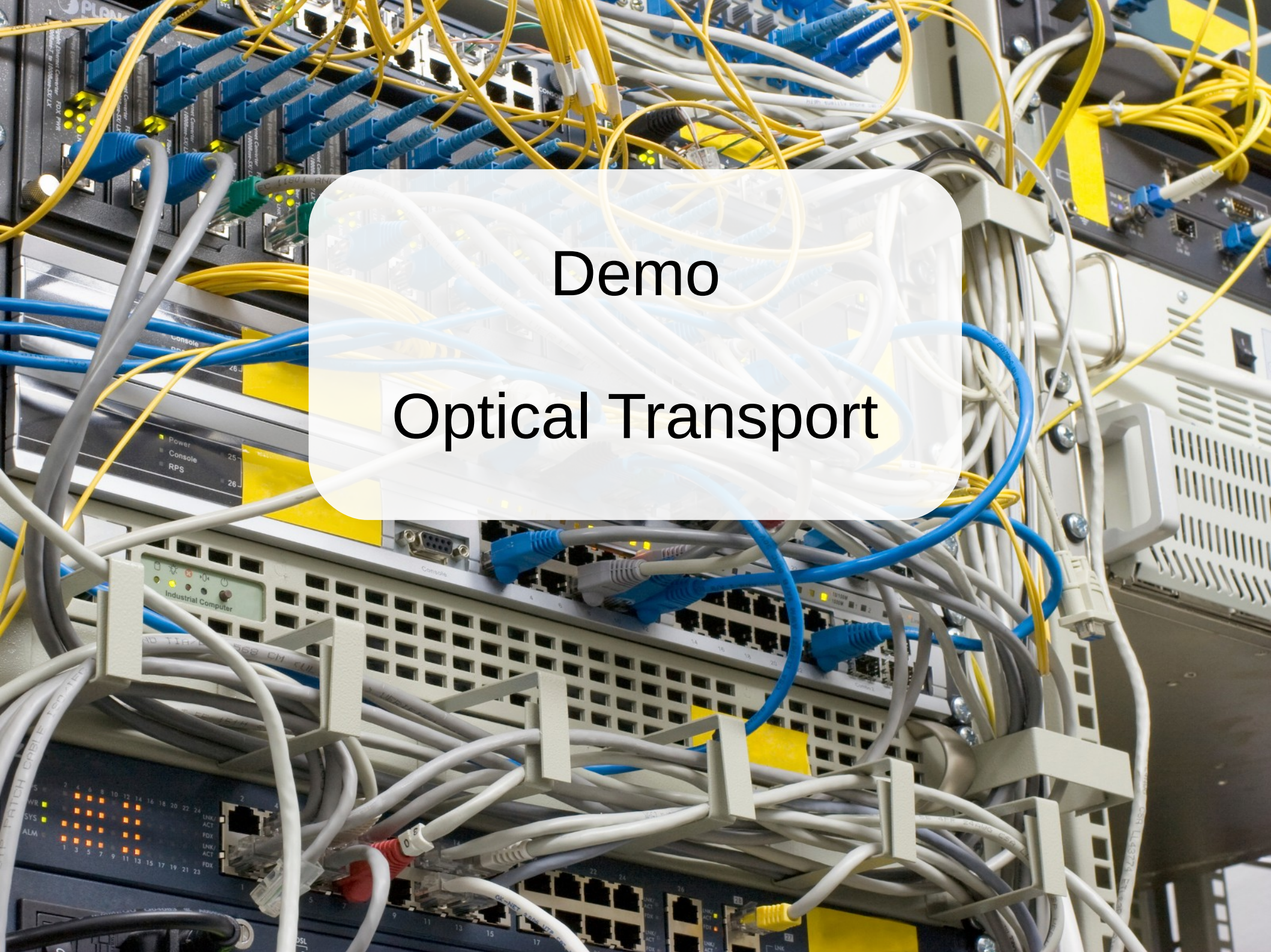
Configuring in the Browser:

Implement configurators in JavaScript.

Configuring in the Browser:

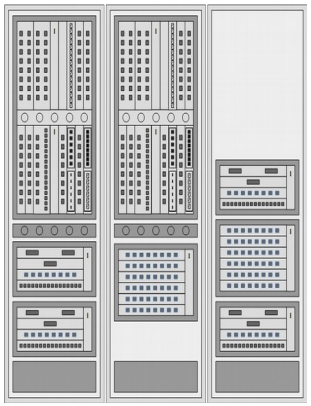
Implement configurators in JavaScript.

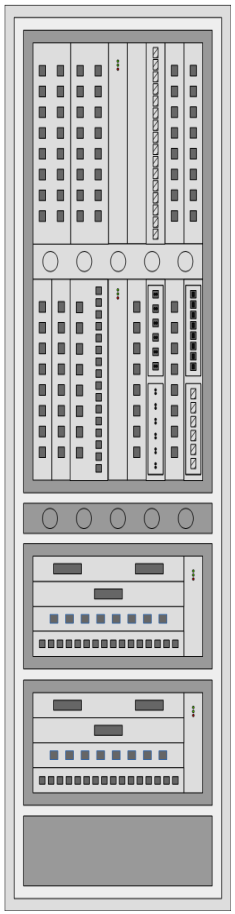
JavaScript is also
a reasonable choice for modeling.

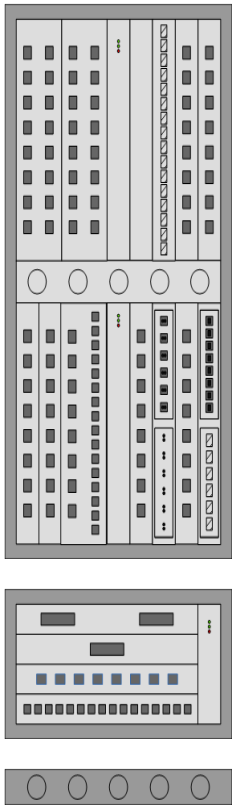

Demo
Optical Transport

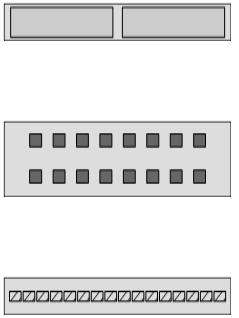
Demo Example: Hierarchical Configuration

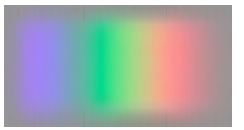
Solution Rack Switch Board (Module) Transceiver (Wavelength)





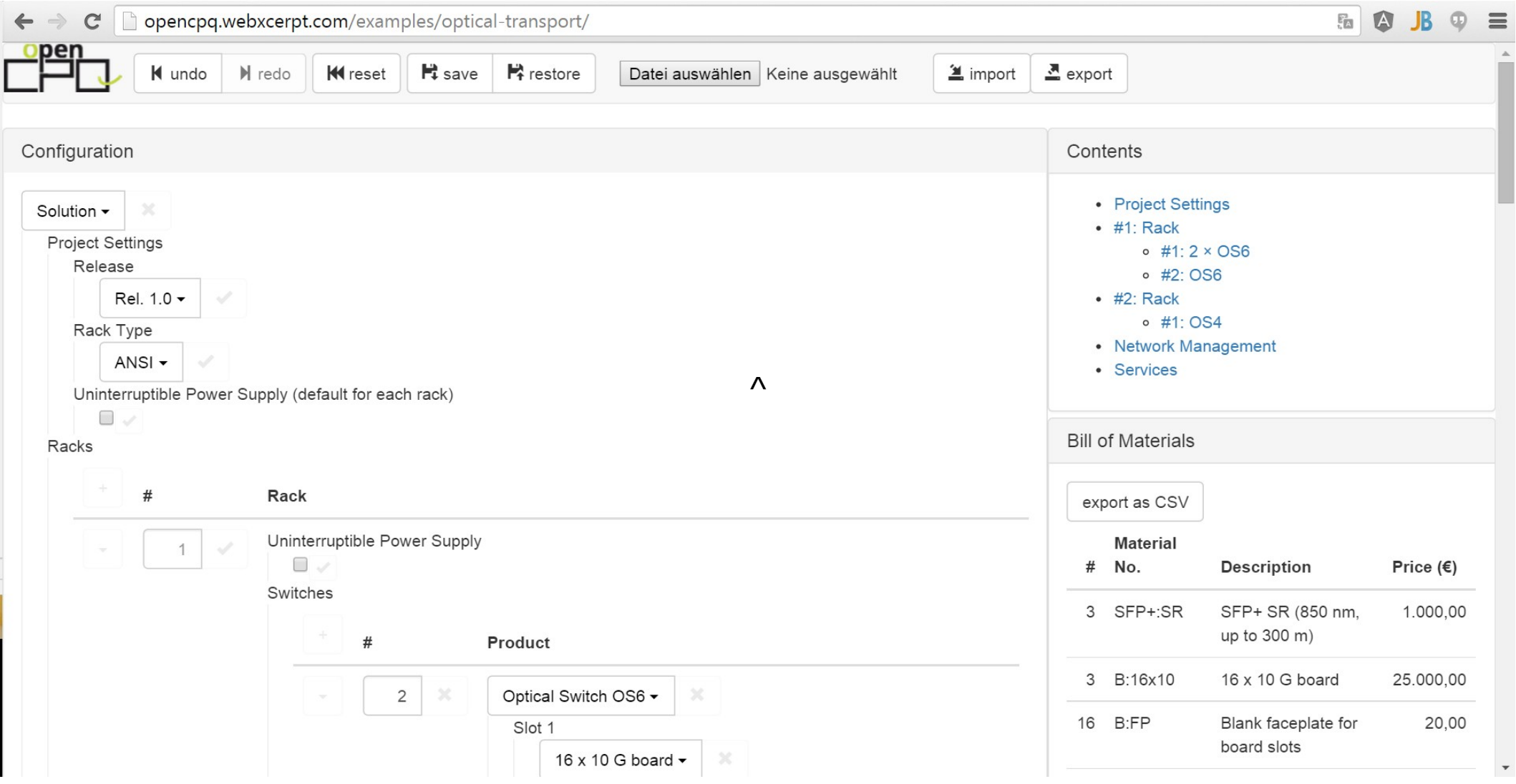


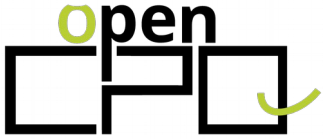




Demo

http://opencpq.webxcerpt.com/examples/optical-transport/

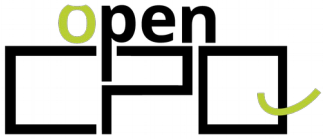




– a JavaScript-based Configurator Framework

– a JavaScript-based Configurator Framework

- Building-block library
  - Components
  - Dependencies



– a JavaScript-based Configurator Framework

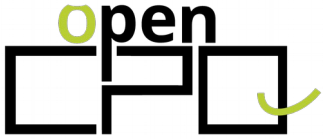
- Building-block library
  - Components
  - Dependencies
- Combine building blocks with JavaScript

open – a JavaScript-based Configurator Framework

- Building-block library
  - Components
  - Dependencies
- Combine building blocks with JavaScript
- Add application-specific building blocks

open – a JavaScript-based Configurator Framework

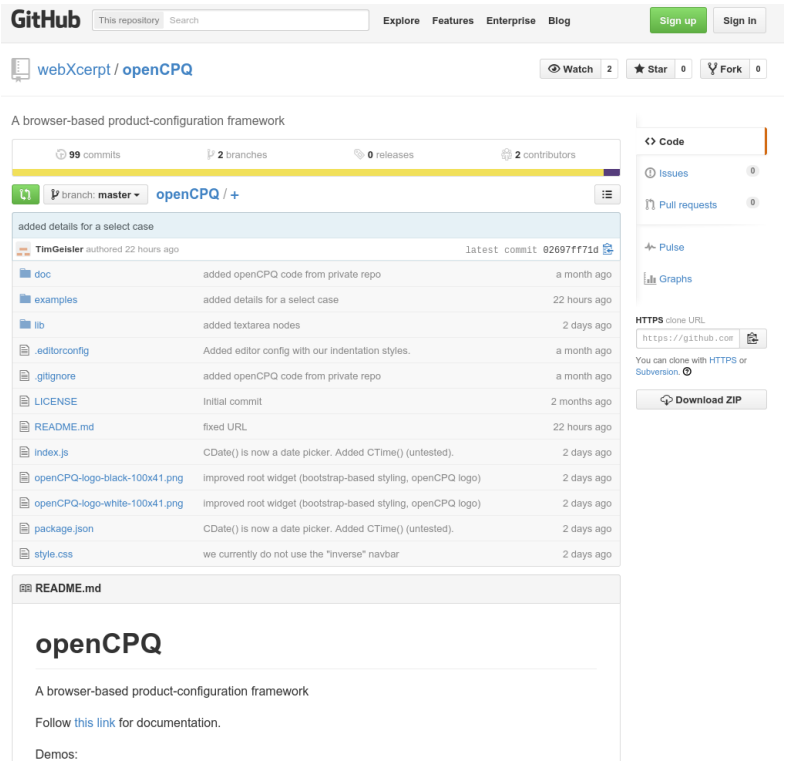
- Building-block library
  - Components
  - Dependencies
- Combine building blocks with JavaScript
- Add application-specific building blocks
- A light-weight layer based on ReactJS and Bootstrap



– an Open-Source Project

– an Open-Source Project

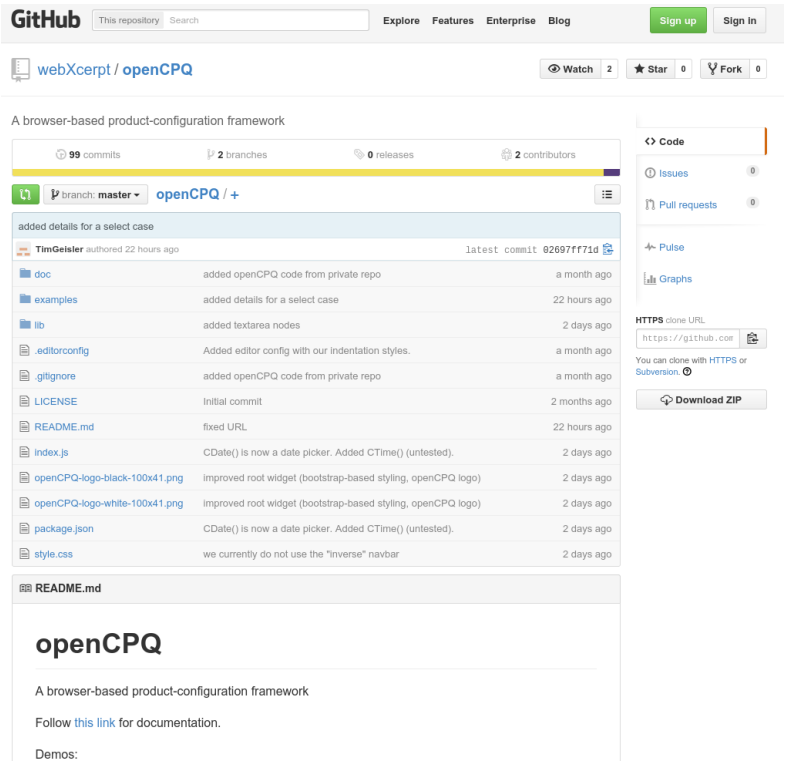
Source code and links to live demos available on Github:
https://github.com/webXcerpt/openCPQ



– an Open-Source Project

Source code and links to live demos available on Github:
https://github.com/webXcerpt/openCPQ

Liberal MIT license

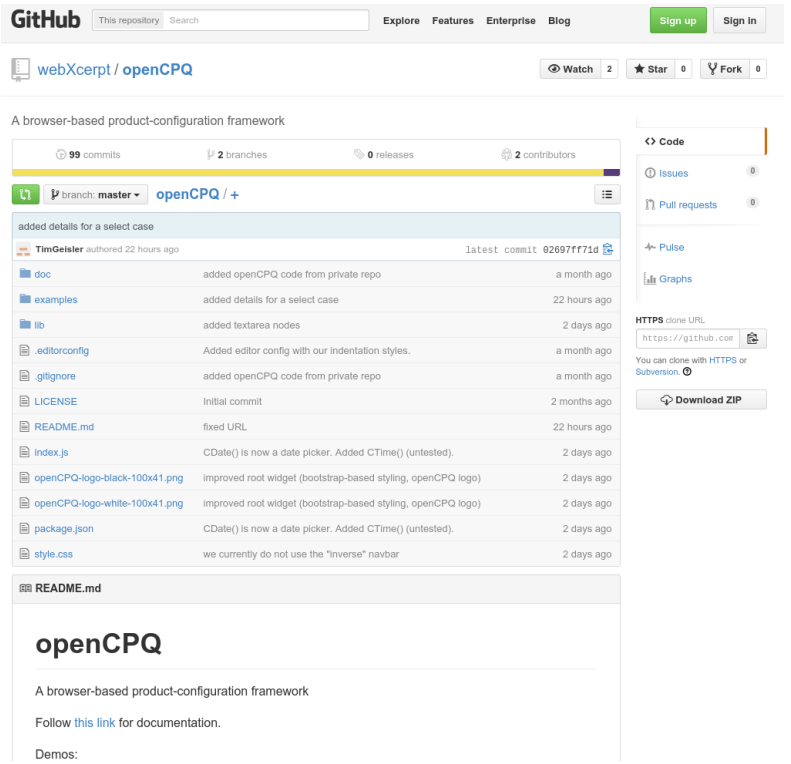


– an Open-Source Project

Source code and links to live demos available on Github:
https://github.com/webXcerpt/openCPQ

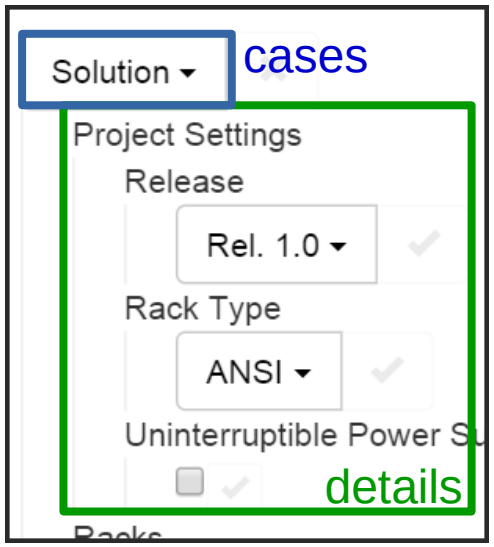
Liberal MIT license

Use, adapt,
integrate, contribute!



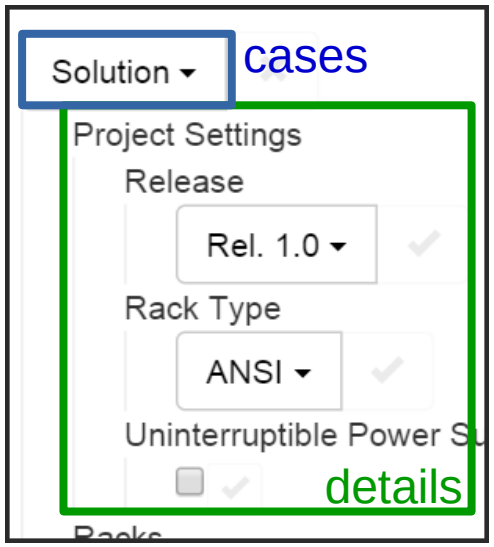
Modeling with openCPQ: Cases with Details

Modeling with openCPQ: Cases with Details

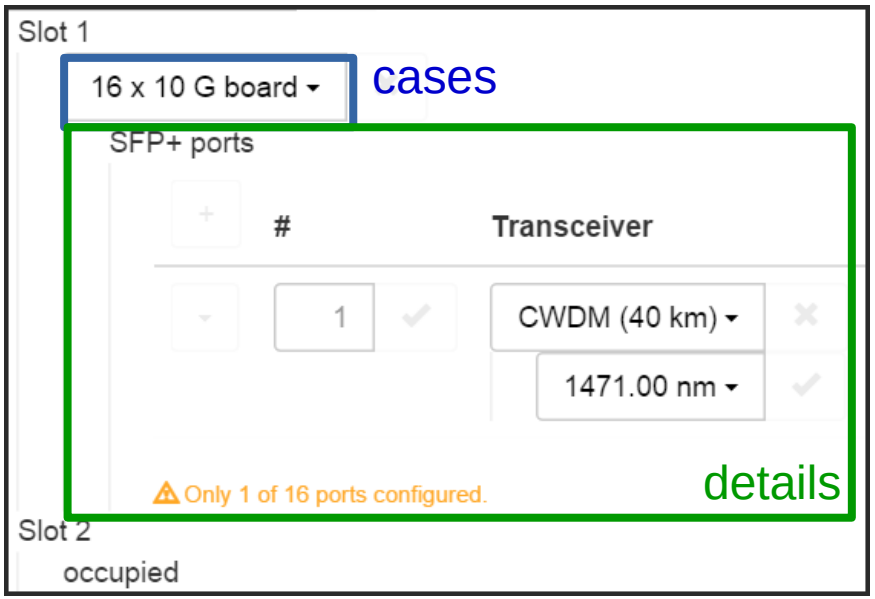



```
var configuration = CSelect([
    unansweredCase("Select Configuration Mode"),
    ccase("Switches", "Optical Switches",
        CQuantifiedList({}, "Optical Switch",
                        opticalSwitches)),
    ccase("Rack",     "Racks",
        CQuantifiedList({}, "Rack",
                        rack)),
    ccase("Solution", "Solution",
          solution),
]);
```

Modeling with openCPQ: Cases with Details




```
var configuration = CSelect([
    unansweredCase("Select Configuration Mode"),
    ccase("Switches", "Optical Switches",
        CQuantifiedList({}, "Optical Switch",
                        opticalSwitches)),
    ccase("Rack",     "Racks",
        CQuantifiedList({}, "Rack",
                        rack)),
    ccase("Solution", "Solution",
          solution),
]);
```




```
function boards(isDoubleWidthSlot) {
    return CSelect([
        for (b of components.boards)
            if (!b.doubleWidth ||
                isDoubleWidthSlot)
                ccaseBOM(b.name, b.label,
                    ports(b.ports))
    ]);
}
```

Data-Driven Modeling with openCPQ

Data-Driven Modeling with openCPQ

Boards

Name	Label	Double width	Power	Ports		
				Label	Count	Type
B:FP	unequipped					
B:8x10_16x1	8 x 10 G + 16 x 1 G board	y	45	SFP+ ports	8	SFP+
				SFP ports	16	SFP
B:8x10	8 x 10 G board		30	SFP+ ports	8	SFP+
B:16x10	16 x 10 G board	y	50	SFP+ ports	16	SFP+
B:16xE1_75	16 x E1 electrical board (75 Ohm)		40			
B:16xE1_120	16 x E1 electrical board (120 Ohm)		40			
B:2x40	2 x 40 G board		60	QSFP+ ports	2	QSFP+
B:1x100	1 x 100 G board		60	CFP ports	1	CFP

Data-Driven Modeling with openCPQ

Boards

Name	Label	Double width	Power	Ports		
				Label	Count	Type
B:FP	unequipped					
B:8x10_16x1	8 x 10 G + 16 x 1 G board	y	45	SFP+ ports	8	SFP+
				SFP ports	16	SFP
B:8x10	8 x 10 G board		30	SFP+ ports	8	SFP+
B:16x10	16 x 10 G board	y	50	SFP+ ports	16	SFP+
B:16xE1_75	16 x E1 electrical board (75 Ohm)		40			
B:16xE1_120	16 x E1 electrical board (120 Ohm)		40			
B:2x40	2 x 40 G board		60	QSFP+ ports	2	QSFP+
B:1x100	1 x 100 G board		60	CFP ports	1	CFP

Data-Driven Modeling with openCPQ

Boards

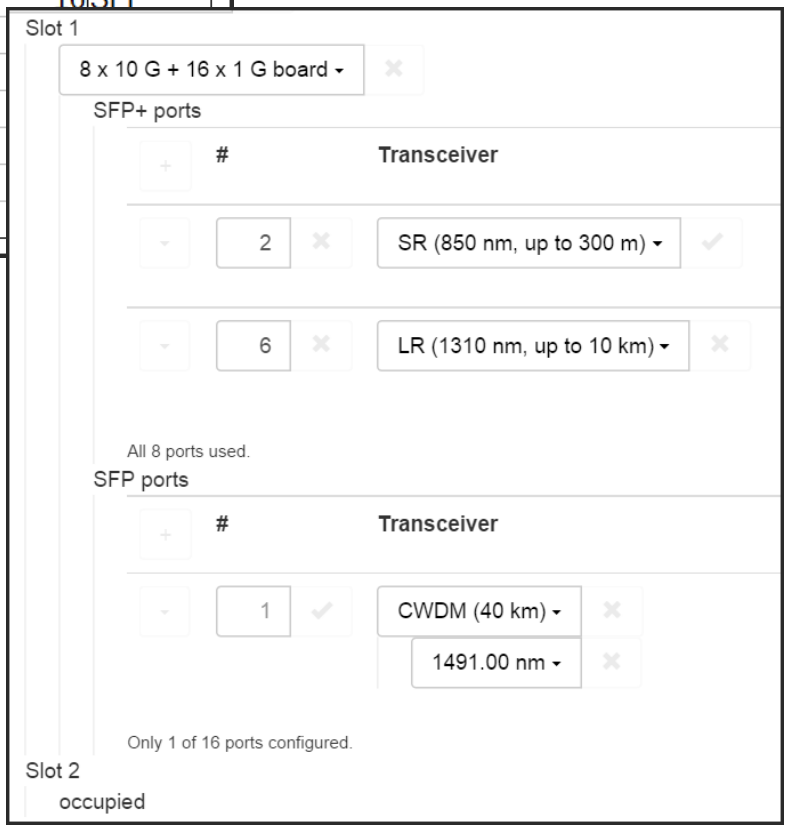
Name	Label	Double width	Power	Ports		
				Label	Count	Type
B:FP	unequipped					
B:8x10_16x1	8 x 10 G + 16 x 1 G board	y	45	SFP+ ports	8	SFP+
				SFP ports	16	SFP
B:8x10	8 x 10 G board		30	SFP+ ports		
B:16x10	16 x 10 G board	y	50	SFP+ ports		
B:16xE1_75	16 x E1 electrical board (75 Ohm)		40			
B:16xE1_120	16 x E1 electrical board (120 Ohm)		40			
B:2x40	2 x 40 G board		60	QSFP+ ports		
B:1x100	1 x 100 G board		60	CFP ports		

Data-Driven Modeling with openCPQ

Boards

Name	Label	Double width	Power	Ports		
				Label	Count	Type
B:FP	unequipped					
B:8x10_16x1	8 x 10 G + 16 x 1 G board	y	45	SFP+ ports	8	SFP+
				SFP ports	16	SFP
B:8x10	8 x 10 G board		30	SFP+ ports		
B:16x10	16 x 10 G board	y	50	SFP+ ports		
B:16xE1_75	16 x E1 electrical board (75 Ohm)		40			
B:16xE1_120	16 x E1 electrical board (120 Ohm)		40			
B:2x40	2 x 40 G board		60	QSFP+ ports		
B:1x100	1 x 100 G board		60	CFP ports		

```
function boards(isDoubleWidthSlot) {
    return CSelect([
        for (b of components.boards)
            if (!b.doubleWidth || isDoubleWidthSlot)
                ccaseBOM(b.name, b.label,
                    aggregate("power", b.power,
                        ports(b.ports)))
    ]);
}
```



Data-Driven Modeling with openCPQ

Boards

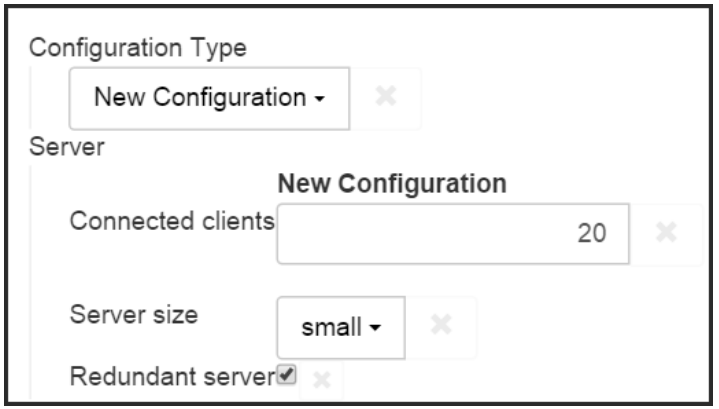
Name	Label	Double width	Power	Ports		
				Label	Count	Type
B:FP	unequipped					
B:8x10_16x1	8 x 10 G + 16 x 1 G board	y	45	SFP+ ports	8	SFP+
				SFP ports	16	SFP
B:8x10	8 x 10 G board		30	SFP+ ports		
B:16x10	16 x 10 G board	y	50	SFP+ ports		
B:16xE1_75	16 x E1 electrical board (75 Ohm)		40			
B:16xE1_120	16 x E1 electrical board (120 Ohm)		40			
B:2x40	2 x 40 G board		60	QSFP+ ports		
B:1x100	1 x 100 G board		60	CFP ports		


```
function boards(isDoubleWidthSlot) {
    return CSelect([
        for (b of components.boards)
            if (!b.doubleWidth || isDoubleWidthSlot)
                ccaseBOM(b.name, b.label,
                    aggregate("power", b.power,
                        ports(b.ports)))
    ]);
}
```

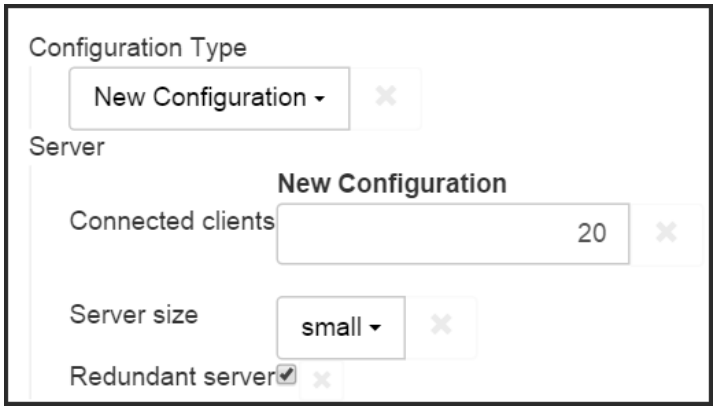
➡ Concise specification of complex models

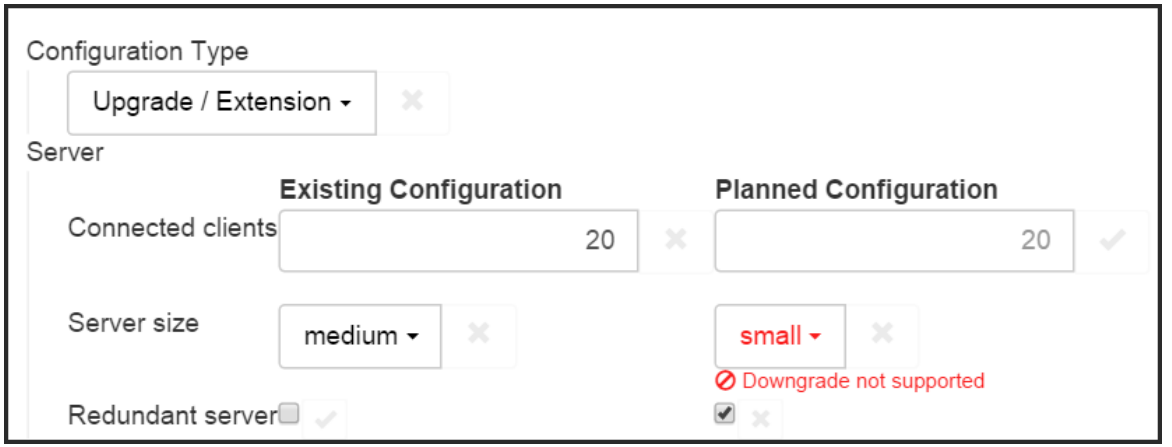
Modeling with openCPQ: Application-specific Abstractions

Modeling with openCPQ: Application-specific Abstractions

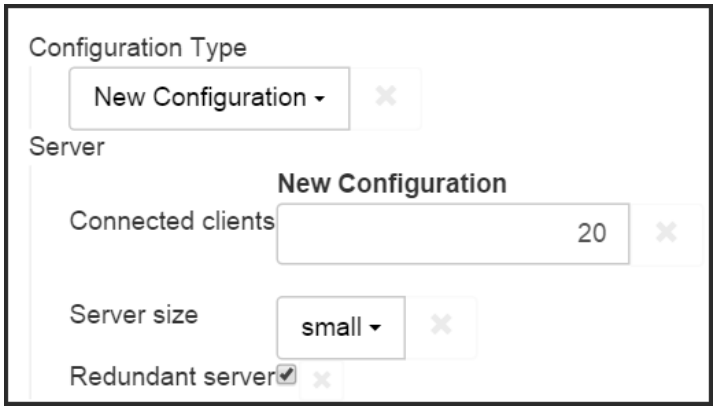


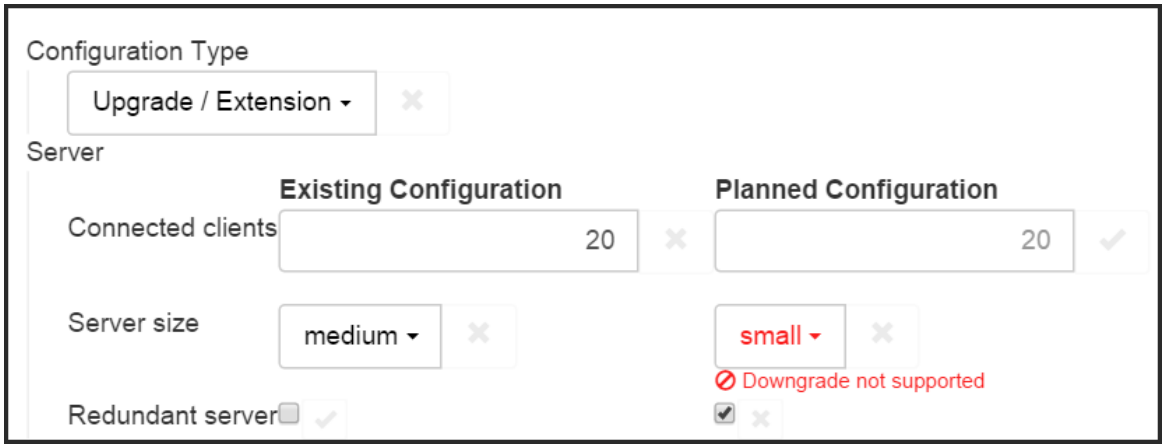
Modeling with openCPQ: Application-specific Abstractions





Modeling with openCPQ: Application-specific Abstractions



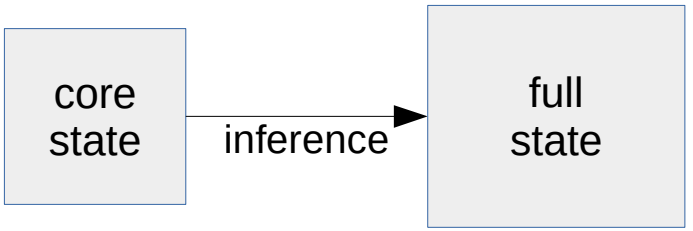



```
CNameSpace("props", CGroup([
    cmember("ConfigType", "Configuration Type",
        CNamed("props", "ConfigType", {valueAccessor: n => n.value}, CSelect([
            ccase("NEW", "New Configuration"), ccase("EXT", "Upgrade / Extension"),
    ]))),
    cmember("Server", "Server", ep.table([
        ep.rowInteger("clients", "Connected clients"),
        crow("Size", "Server size", ({props}) => props.ConfigType === "EXT"
            ? [ep.eCell("Size", CSelect([for (s of serverSizes) ccase(s)])),
               () => ep.pCell("Size", CSelect([for (s of serverSizes)
                   onlyIf(serverSizes.indexOf(s) >= serverSizes.indexOf(ep.E(props.Size)),
                          "Downgrade not supported", [ccase(s)])]))]
            : [ep.pCell("Size", CSelect([for (s of serverSizes) ccase(s)]))]
        ),
        ep.rowBoolean("redundancy", "Redundant server"),
    ])),
])),
```

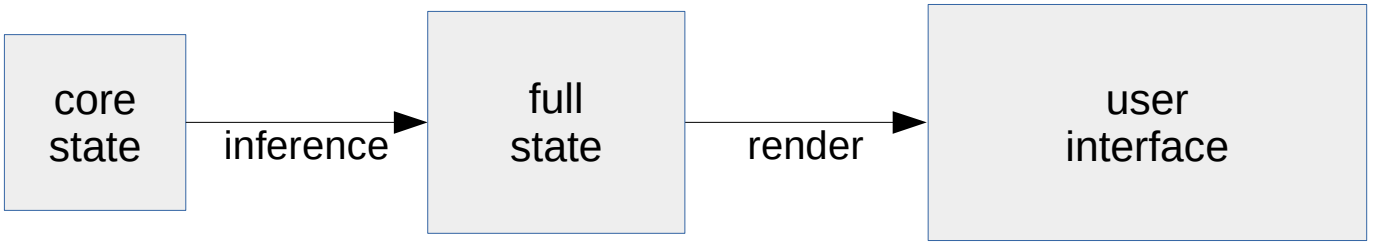
How it Works

Change Propagation

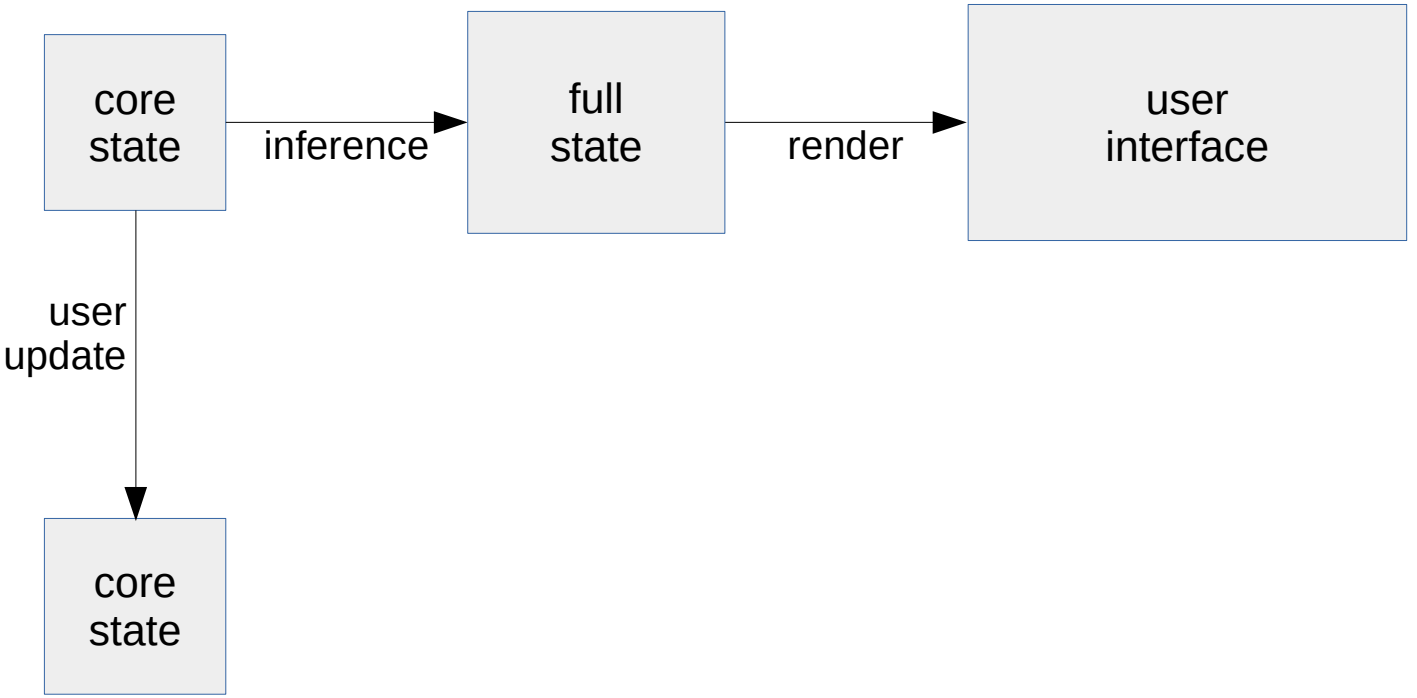
Change Propagation



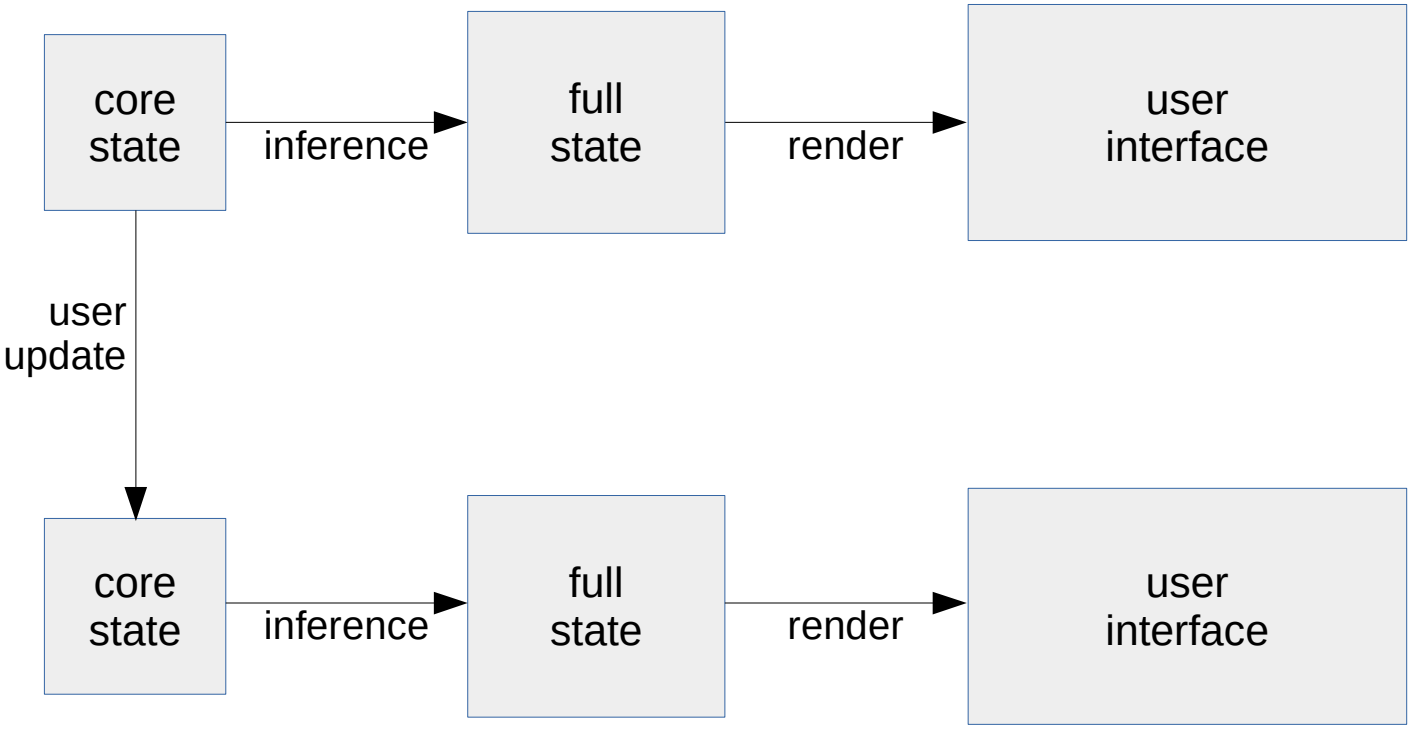
Change Propagation



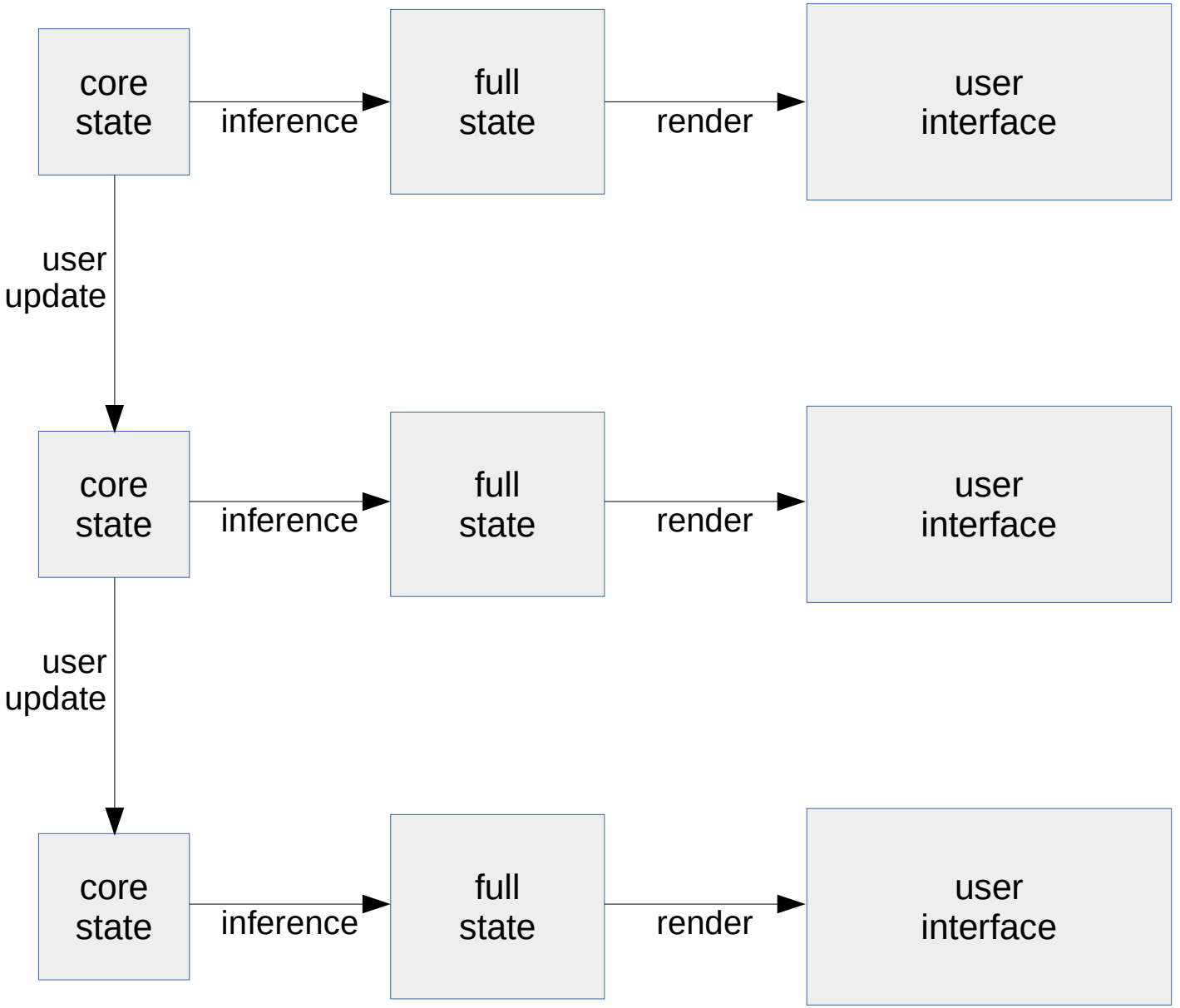
Change Propagation



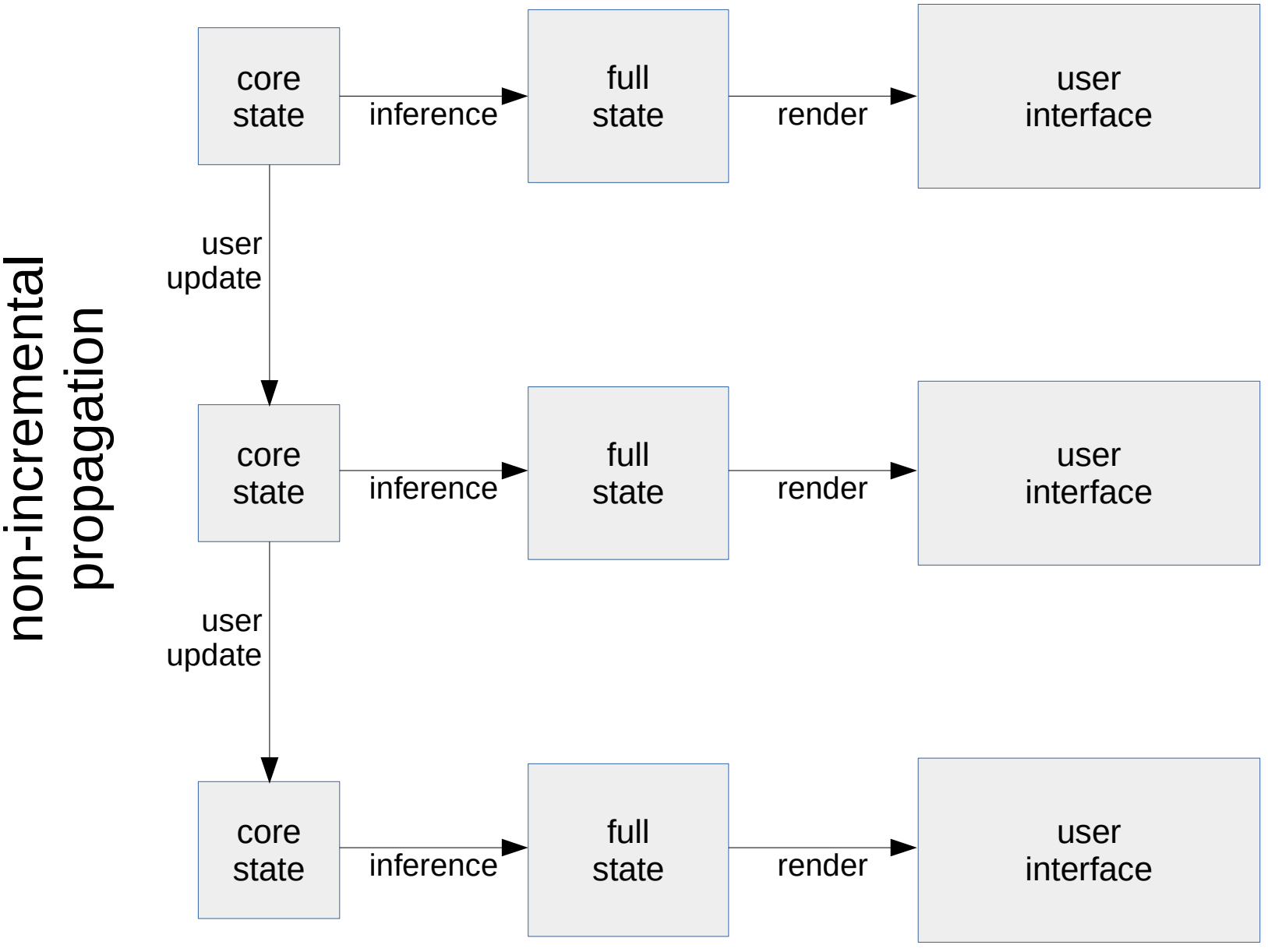
Change Propagation



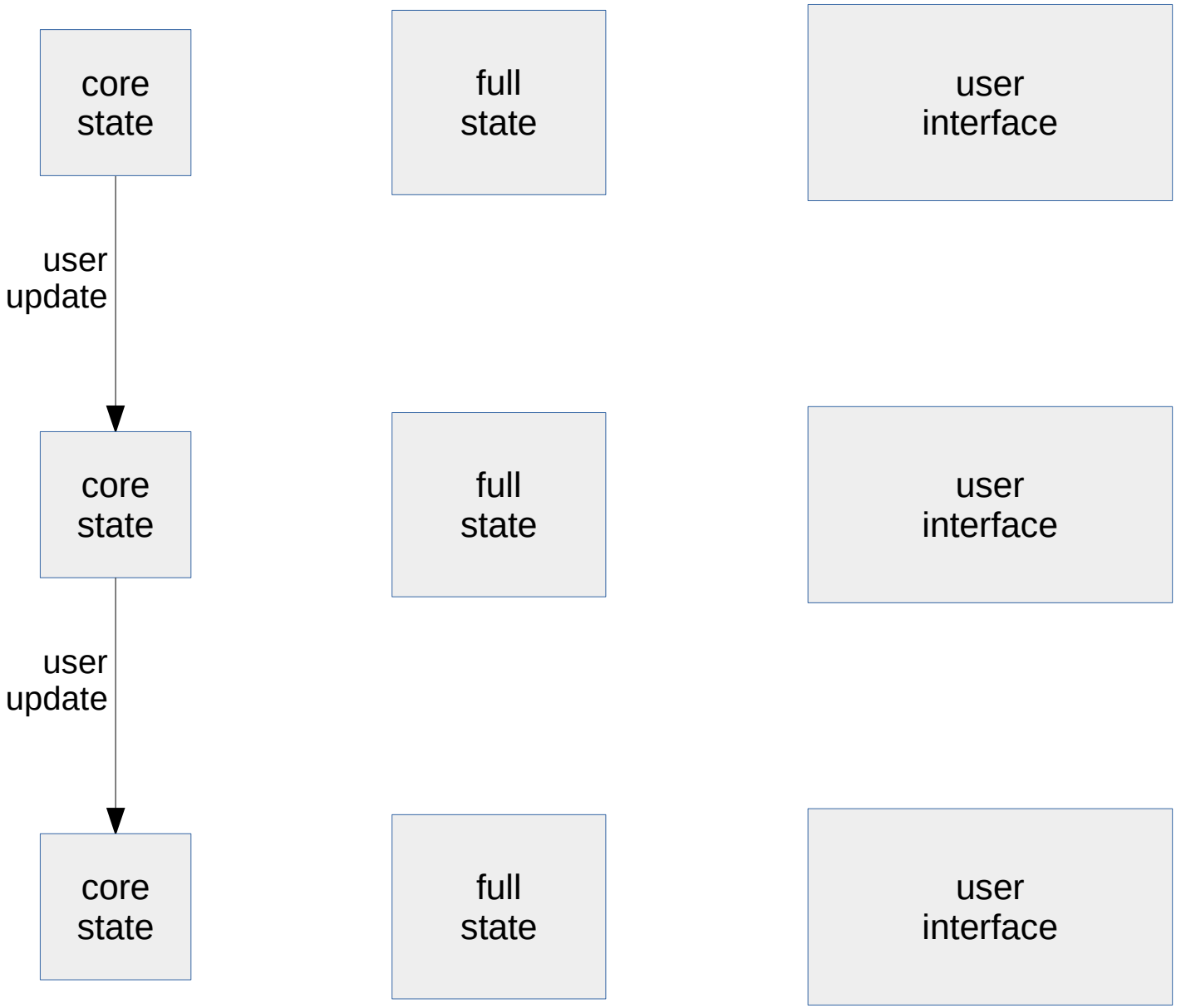
Change Propagation



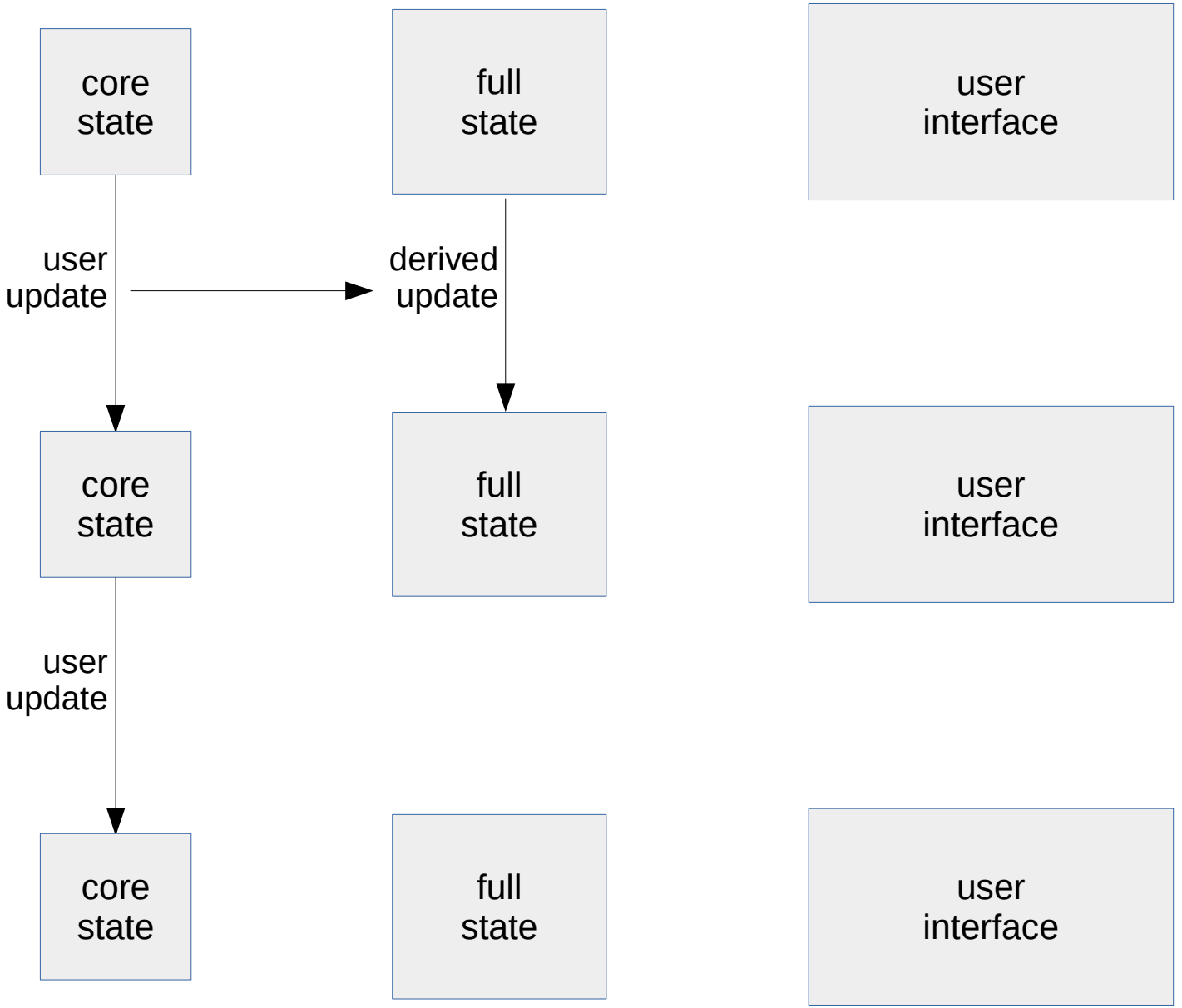
Change Propagation



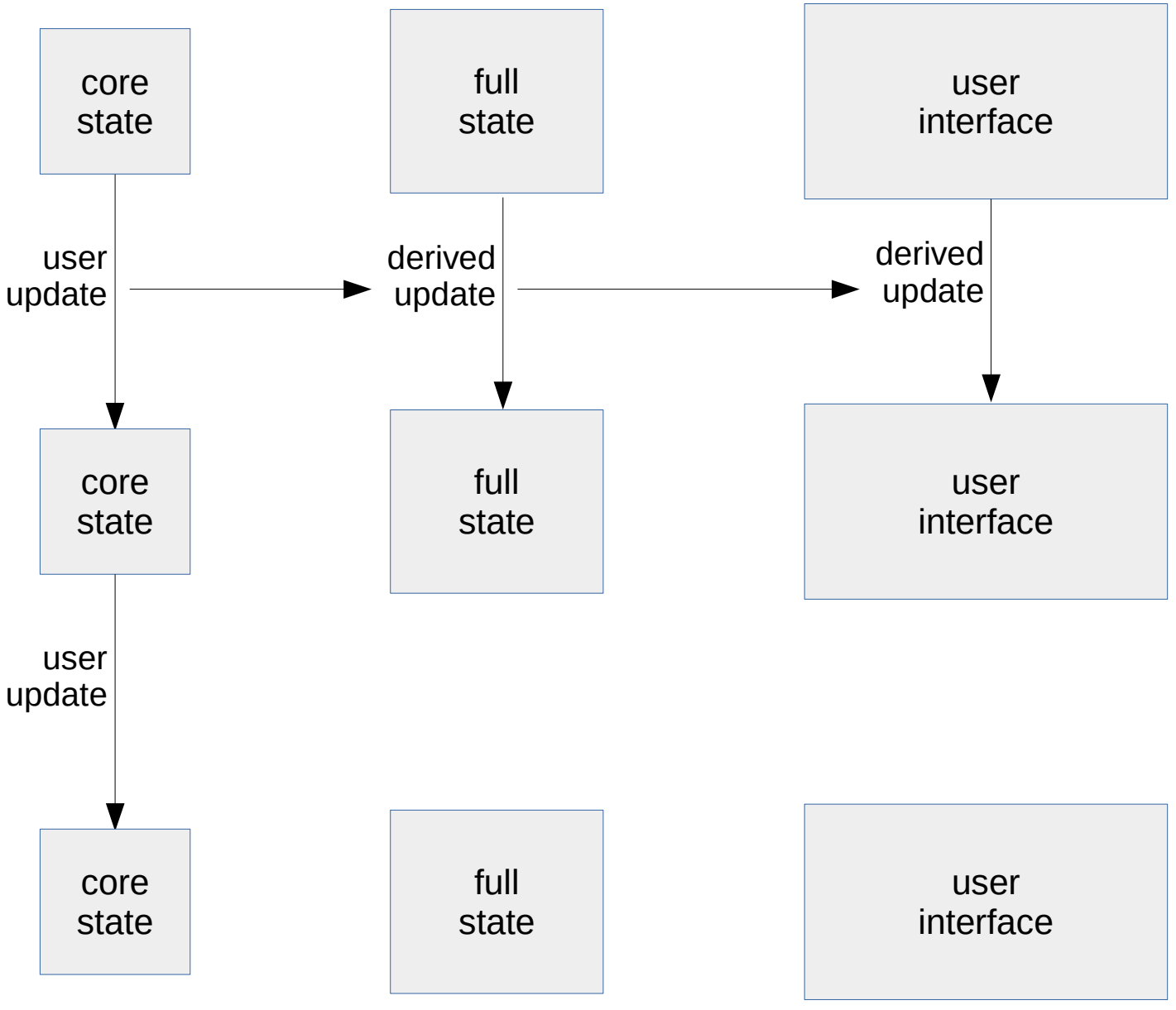
Change Propagation



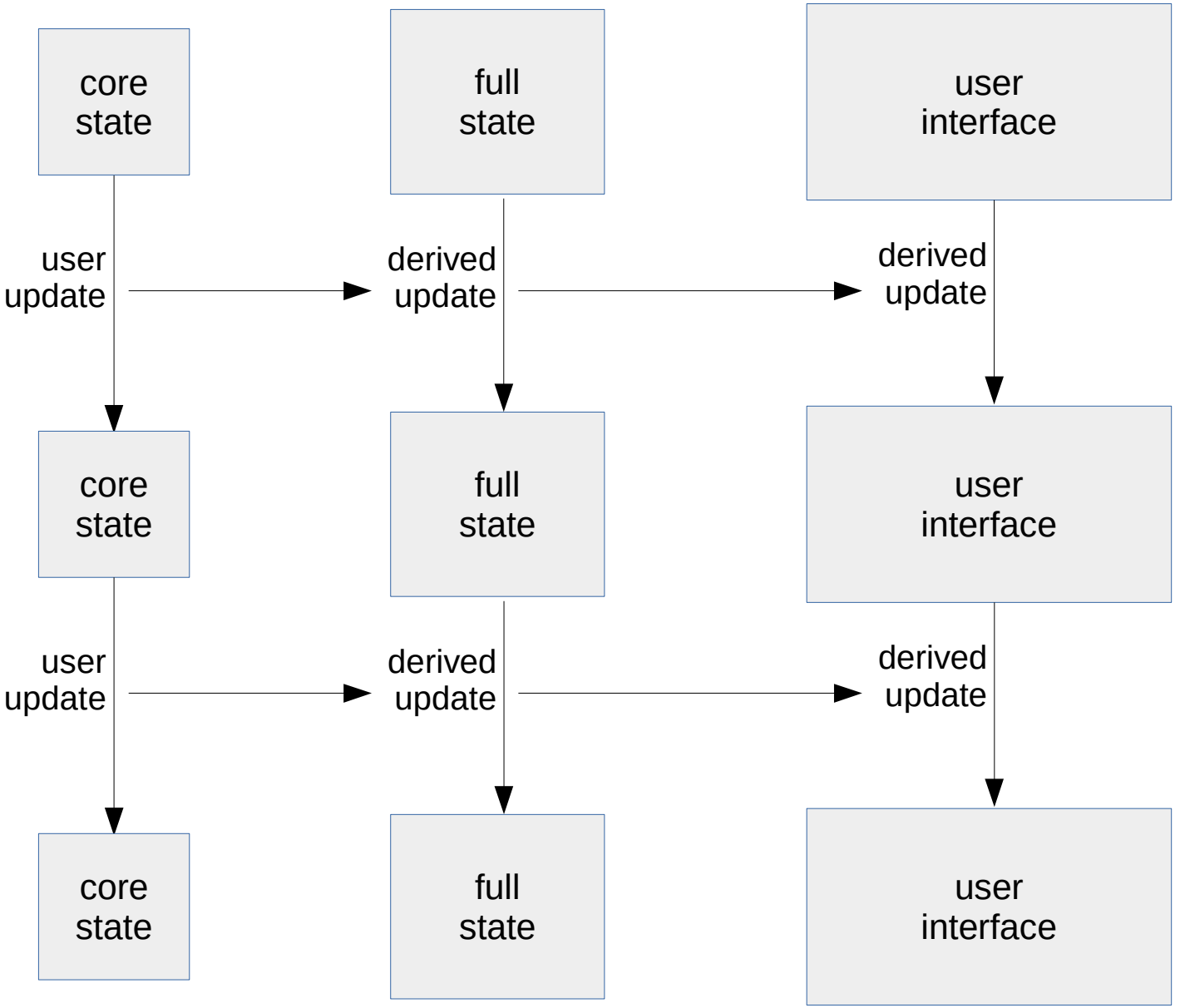
Change Propagation



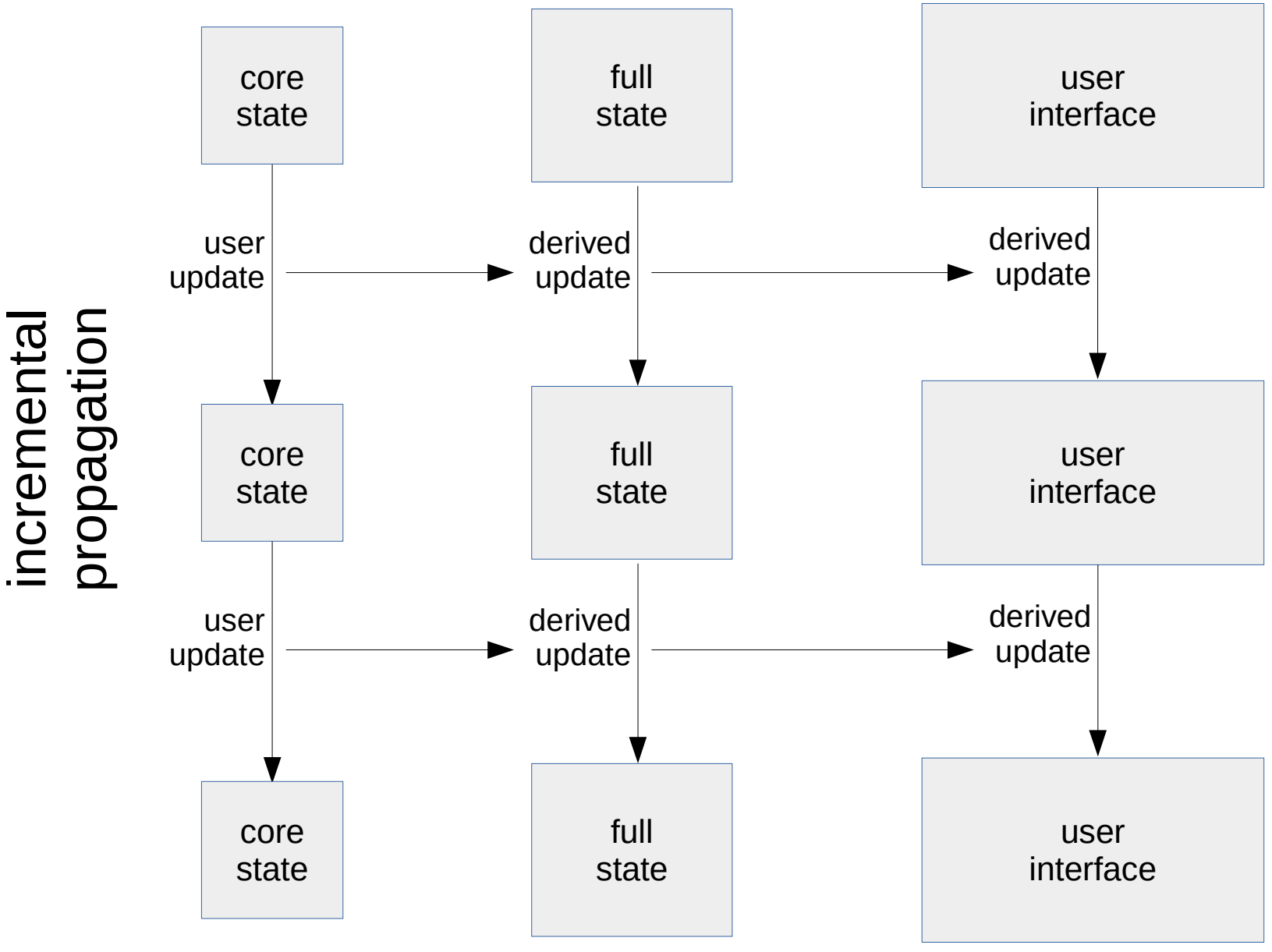
Change Propagation



Change Propagation



Change Propagation



Change Propagation

Change Propagation

Trade-off:

- Non-incremental propagation:

Change Propagation

Trade-off:

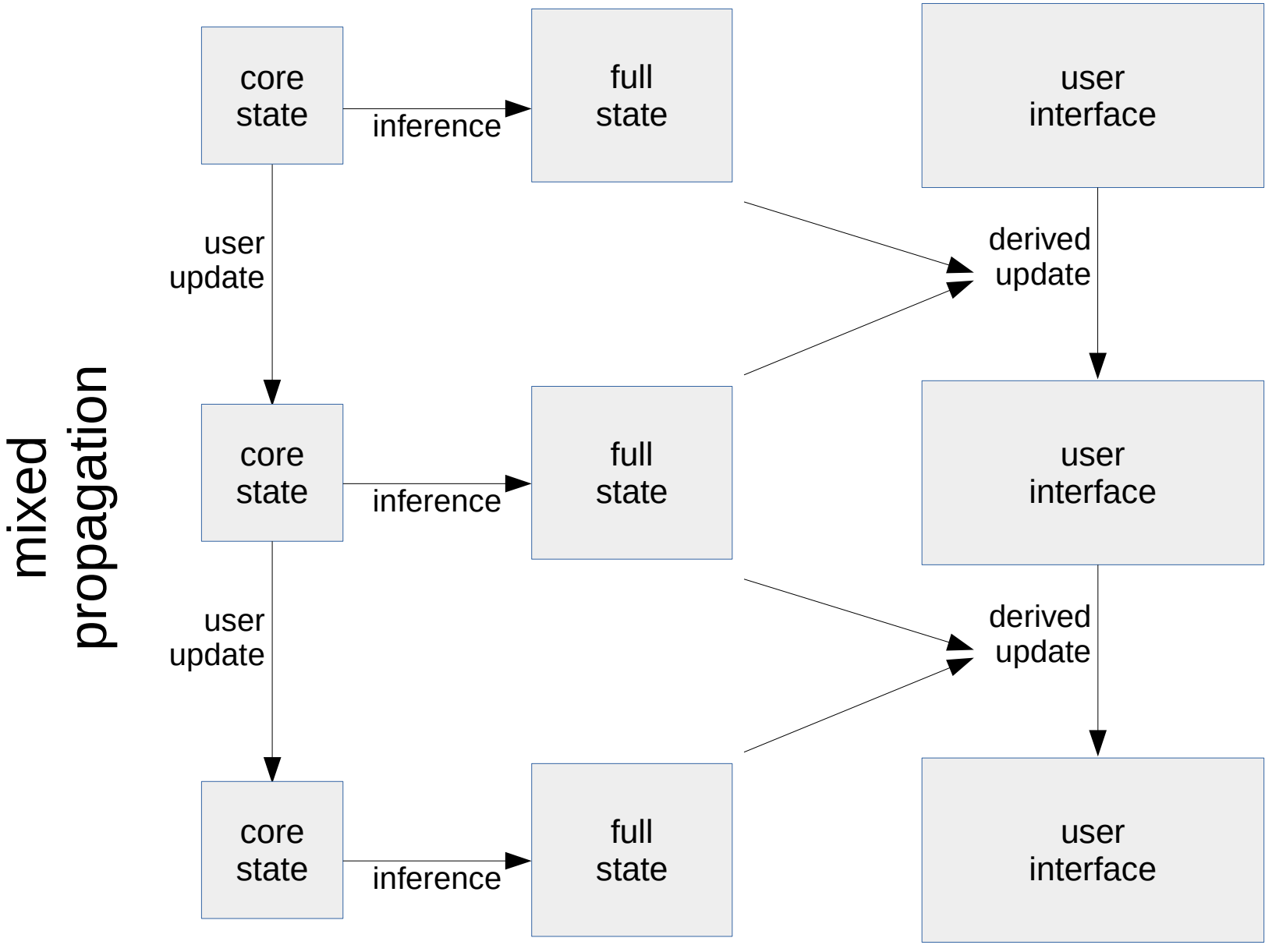
- Non-incremental propagation:
  - Redo inference steps
    - CPU consumption
  - Redo rendering
    - CPU consumption
    - Flicker, loss of UI state (focus, scroll, selection), ...
- Incremental propagation:

Change Propagation

Trade-off:

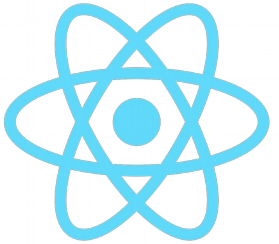
- Non-incremental propagation:
  - Redo inference steps
    - CPU consumption
  - Redo rendering
    - CPU consumption
    - Flicker, loss of UI state (focus, scroll, selection), ...
- Incremental propagation:
  - Keep track of dependencies
    - Error-prone (unless completely shielded from the modeler)
    - Consumes memory and CPU

Change Propagation



React:

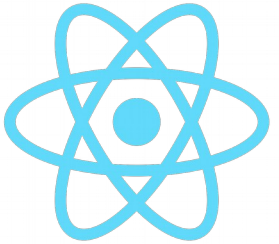
A JavaScript library for building user interfaces



React:

A JavaScript library for building user interfaces

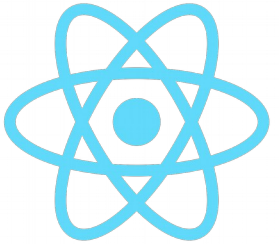
- Unique approach:
  - not a widget library
  - not an MVC framework



React:

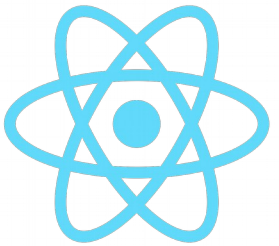
A JavaScript library for building user interfaces

- Unique approach:
  - not a widget library
  - not an MVC framework
- Virtual DOM ("VDOM"):
  - Representation of the DOM tree as a JavaScript data structure (cheap!)



React:

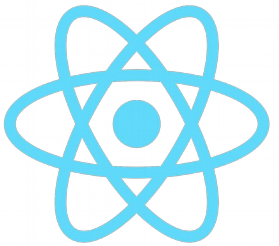
A JavaScript library for building user interfaces



- Unique approach:
  - not a widget library
  - not an MVC framework
- Virtual DOM ("VDOM"):
  - Representation of the DOM tree as a JavaScript data structure (cheap!)
- Upon each update:

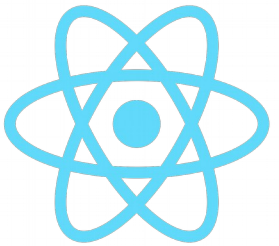
React:

A JavaScript library for building user interfaces



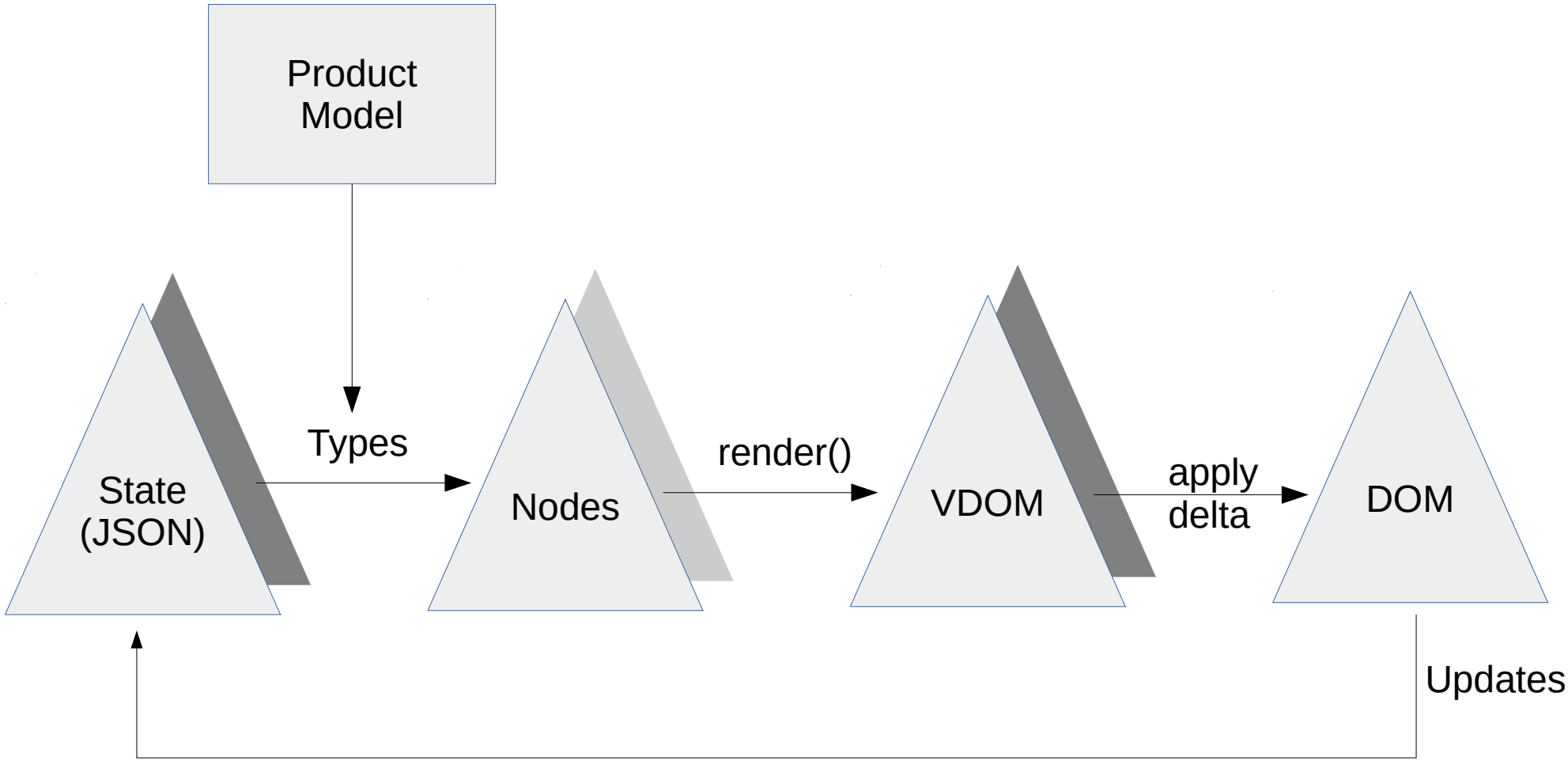
- Unique approach:
  - not a widget library
  - not an MVC framework
- Virtual DOM ("VDOM"):
  - Representation of the DOM tree as a JavaScript data structure (cheap!)
- Upon each update:
  - User code
    - generates VDOM from your model
    - possibly using XML templating integrated into JavaScript ("JSX")

React:
A JavaScript library for building user interfaces

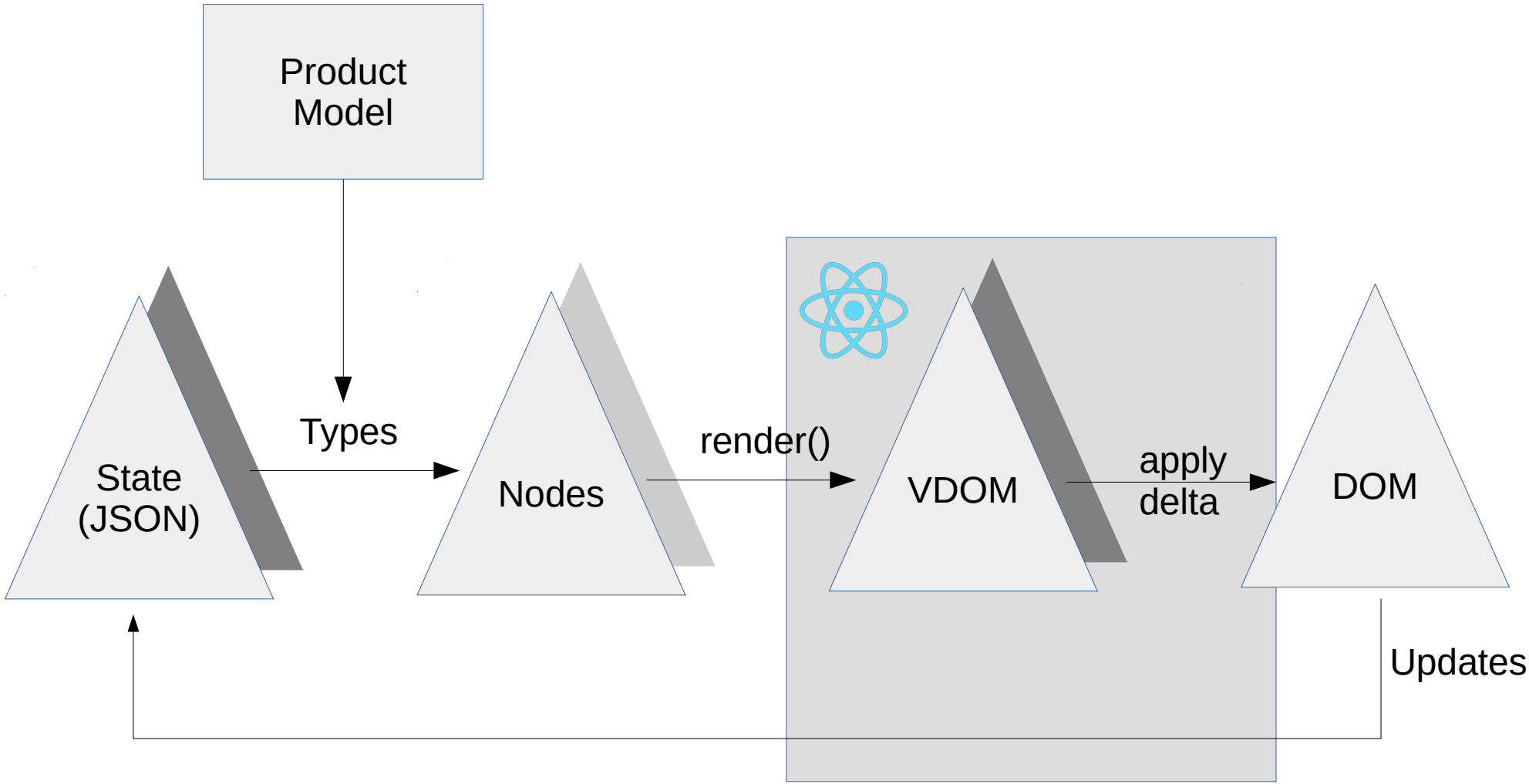


- Unique approach:
  - not a widget library
  - not an MVC framework
- Virtual DOM ("VDOM"):
  - Representation of the DOM tree as a JavaScript data structure (cheap!)
- Upon each update:
  - User code
    - generates VDOM from your model
    - possibly using XML templating integrated into JavaScript ("JSX")
  - React
    - diffs the VDOM with the previous VDOM
    - applies only the diff to the actual DOM

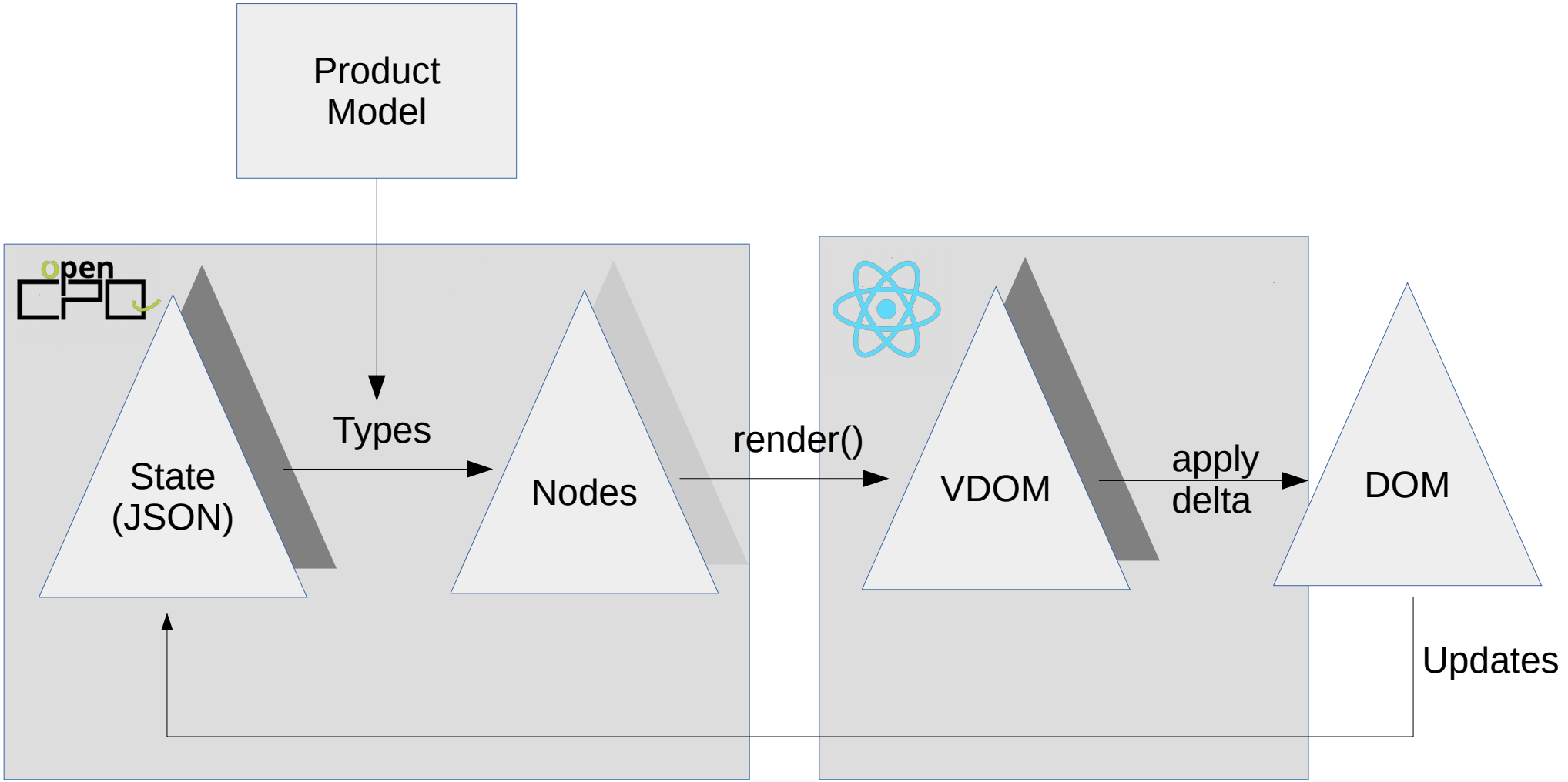
Architecture



Architecture



Architecture



SAP Integration

SAP Integration

- Models
  - Conversion of LO-VC and IPC models to openCPQ
    - Schema, basic logic: automatable with VClipse extension
    - Complex logic: manual conversion

SAP Integration

- Models
  - Conversion of LO-VC and IPC models to openCPQ
    - Schema, basic logic: automatable with VClipse extension
    - Complex logic: manual conversion
  - Model storage and management
    - Just static resources
    - App server not needed (but can be used)

SAP Integration

SAP Integration

- Data
  (e.g. materials with classification information)
  - Live vs. pre-exported
  - Bundling with application vs. loading on demand

SAP Integration

- Data
  (e.g. materials with classification information)
  - Live vs. pre-exported
  - Bundling with application vs. loading on demand
- Runtime
  - Loading and saving configurations
    - External configurator API
    - Mimic IPC

Summary

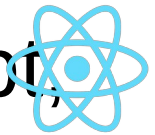
Summary

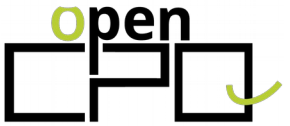
Take advantage of modern **browser technology** for product configuration.

Summary

Take advantage of modern **browser technology** for product configuration.

Powerful **modeling** based on JavaScript, React, and openCPQ.

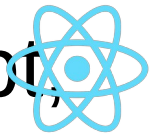


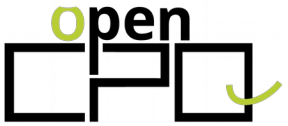


Summary

Take advantage of modern **browser technology** for product configuration.

Powerful **modeling** based on JavaScript, React, and openCPQ.



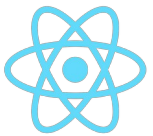
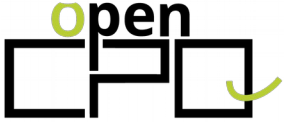


Flexible and fast **user interface**.

Summary

Take advantage of modern **browser technology** for product configuration.


Powerful **modeling** based on JavaScript, React, and openCPQ.

Flexible and fast **user interface**.

Use, adapt, integrate, contribute!
https://github.com/webXcerpt/openCPQ

Our Offer

Our Offer

Discuss:

- Use cases, modeling challenges, …
- Integrations

Our Offer

Discuss:

- Use cases, modeling challenges, …
- Integrations

Cooperate:

- Professional services, training, ...
- For end users or integrators