Front-end architecture in practice 🔧

#whoami

Charly POLY - Sr Software Engineer at ⏱ algolia

Past

➡️ ex-Dashlane

➡️ ex-JobTeaser

Why talk about "Front-end architecture"?

➡️ Because the web has evolved!

Why talk about "Front-end architecture"?

➡️ Because the web has evolved!

1

00'

Why talk about "Front-end architecture"?

➡️ Because the web has evolved!

1

2

00'

10'

Why talk about "Front-end architecture"?

➡️ Because the web has evolved!

1	2	3
00'	10'	20'

The Front-end evolution

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
How text is interpreted when uninfluenced by tags
The tags
A list of the tags used in HTML with their significance.
Entities
Special characters are represented by SGML entities
DTD
The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
Changes suggested for HTML improvements
Constraints
Design constraints for HTML which might explain some of its properties.

See also

New spec
As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.

```
$ telnet www.perdu.com 80              Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                         Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                        Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
                                       Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
    How text is interpreted when uninfluenced by tags
The tags
    A list of the tags used in HTML with their significance.
Entities
    Special characters are represented by SGML entities
DTD
    The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
    A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
    Changes suggested for HTML improvements
Constraints
    Design constraints for HTML which might explain some of its properties.

See also

New spec
    As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.

```
$ telnet www.perdu.com 80                    Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                               Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                              Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>       Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb
- 91: HTML arrive

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW parsers should ignore tags which they do not understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
    How text is interpreted when uninfluenced by tags
The tags
    A list of the tags used in HTML with their significance.
Entities
    Special characters are represented by SGML entities
DTD
    The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
    A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
    Changes suggested for HTML improvements
Constraints
    Design constraints for HTML which might explain some of its properties.

See also

New spec
    As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.

```
$ telnet www.perdu.com 80                    Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                               Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                              Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>       Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb

- 91: HTML arrive

- 93': CGI scripts

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
    How text is interpreted when uninfluenced by tags
The tags
    A list of the tags used in HTML with their significance.
Entities
    Special characters are represented by SGML entities
DTD
    The SGML document type definition for HTML.

The following do not form part of the specifcation.

Example
    A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
    Changes suggested for HTML improvements
Constraints
    Design constraints for HTML which might explain some of its properties.

See also

New spec
    As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.

```
$ telnet www.perdu.com 80              Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                         Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                        Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>    Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb
- 93': CGI scripts

- 91: HTML arrive
- 94: CSS arrive

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW
understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
    How text is interpreted when uninfluenced by tags
The tags
    A list of the tags used in HTML with their significance.
Entities
    Special characters are represented by SGML entities
DTD
    The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
    A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
    Changes suggested for HTML improvements
Constraints
    Design constraints for HTML which might explain some of its properties.

See also

New spec
    As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.

```
$ telnet www.perdu.com 80              Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                         Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                        Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
                                       Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb

- 93': CGI scripts

- 94-95: Netscape, Opera

- 91: HTML arrive

- 94: CSS arrive

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
  How text is interpreted when uninfluenced by tags
The tags
  A list of the tags used in HTML with their significance.
Entities
  Special characters are represented by SGML entities
DTD
  The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
  A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
  Changes suggested for HTML improvements
Constraints
  Design constraints for HTML which might explain some of its properties.

See also

New spec
  As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.

```
$ telnet www.perdu.com 80                    Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                               Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                              Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
                                             Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb
- 93': CGI scripts
- 94-95: Netscape, Opera

- 91: HTML arrive
- 94: CSS arrive
- 1995: Java applets

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
     How text is interpreted when uninfluenced by tags
The tags
     A list of the tags used in HTML with their significance.
Entities
     Special characters are represented by SGML entities
DTD
     The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
     A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
     Changes suggested for HTML improvements
Constraints
     Design constraints for HTML which might explain some of its properties.

See also

New spec
     As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.


```
$ telnet www.perdu.com 80          Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                     Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                    Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
                                   Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb
- 93': CGI scripts
- 94-95: Netscape, Opera
- 95: JavaScript

- 91: HTML arrive
- 94: CSS arrive
- 1995: Java applets

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW understand, and ignore attributes which they do not understand of tags which they understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
How text is interpreted when uninfluenced by tags
The tags
A list of the tags used in HTML with their significance.
Entities
Special characters are represented by SGML entities
DTD
The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
Changes suggested for HTML improvements
Constraints
Design constraints for HTML which might explain some of its properties.

See also

New spec
As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.


```
$ telnet www.perdu.com 80              Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                          Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                         Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>   * <----- vous
&ecirc;tes ici</pre></strong></body></html>
                                        Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb
- 93': CGI scripts
- 94-95: Netscape, Opera
- 95: JavaScript

- 91: HTML arrive
- 94: CSS arrive
- 1995: Java applets
- 1996: Macromedia Flash

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
    How text is interpreted when uninfluenced by tags
The tags
    A list of the tags used in HTML with their significance.
Entities
    Special characters are represented by SGML entities
DTD
    The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
    A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
    Changes suggested for HTML improvements
Constraints
    Design constraints for HTML which might explain some of its properties.

See also

New spec
    As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.

```
$ telnet www.perdu.com 80              Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                          Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                         Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
                                        Réponse du serveur : body
```

The Front-end evolution

- 90: WorldWideWeb

- 93': CGI scripts

- 94-95: Netscape, Opera

- 95: JavaScript

- 96: HTTP

- 91: HTML arrive

- 94: CSS arrive

- 1995: Java applets

- 1996: Macromedia Flash

HTML

The WWW system uses marked-up text to represent a hypertext document for transmision over the network. The hypertext mark-up language is an SGML format. WWW understand, and ignore attributes which they do not understand of tags which they do understand.

To find out how to write HTML, or to write a program to generate it, read the following sections:-

Default text
    How text is interpreted when uninfluenced by tags
The tags
    A list of the tags used in HTML with their significance.
Entities
    Special characters are represented by SGML entities
DTD
    The SGML document type definition for HTML.

The following do not form part of the specifciation.

Example
    A file containing a variety of tags used for test purposes. See also finding examples on the web .
Future directions
    Changes suggested for HTML improvements
Constraints
    Design constraints for HTML which might explain some of its properties.

See also

New spec
    As edited by Dan Connolly, convex. Comments to www-talk@info.cern.ch please.

```
$ telnet www.perdu.com 80          Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                     Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                    Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
                                   Réponse du serveur : body
```

The Front-end evolution

Back-end

Front-end

The Front-end evolution

Back-end   Front-end

- 1997: PHP 3

The Front-end evolution

Back-end   Front-end

- 1997: PHP 3

- 1999: Zend Engine

The Front-end evolution

Back-end

- 1997: PHP 3
- 1999: Zend Engine

Front-end

- 2000: Adobe Flash with ActionScript

The Front-end evolution

Back-end

- 1997: PHP 3
- 1999: Zend Engine

Front-end

- 2000: Adobe Flash with ActionScript
- 2002: XHR

The Front-end evolution

Back-end

- 1997: PHP 3
- 1999: Zend Engine

Front-end

- 2000: Adobe Flash with ActionScript
- 2002: XHR
- 2004: ActionScript 2

The Front-end evolution

Back-end

- 1997: PHP 3
- 1999: Zend Engine
- 2005: Rails

Front-end

- 2000: Adobe Flash with ActionScript
- 2002: XHR
- 2004: ActionScript 2
- 2005: Prototype
- 2005: Script.aculo.us

The Front-end evolution

Back-end

- 1997: PHP 3
- 1999: Zend Engine
- 2005: Rails

Front-end

- 2000: Adobe Flash with ActionScript
- 2002: XHR
- 2004: ActionScript 2
- 2005: Prototype
- 2005: Script.aculo.us
- 2006: JQuery
- 2006: Mootools
- 2006: YUI

The Front-end evolution

Back-end

- 1997: PHP 3
- 1999: Zend Engine
- 2005: Rails

Front-end

- 2000: Adobe Flash with ActionScript
- 2002: XHR
- 2004: ActionScript 2
- 2005: Prototype
- 2005: Script.aculo.us
- 2006: JQuery
- 2006: Mootools
- 2006: YUI
- ~2008: HTML 5

The Front-end revolution

The Front-end revolution

- 2010: Apple end support of Flash on iOS

The Front-end revolution

- 2010: Apple end support of Flash on iOS

- lodash creator bring Backbone.js

The Front-end revolution

- 2010: Apple end support of Flash on iOS

- lodash creator bring Backbone.js

- Model–view–presenter (MVP) on front side,

  front-end handle its own state

The Front-end revolution

- 2010: Apple end support of Flash on iOS

- lodash creator bring Backbone.js

- Model–view–presenter (MVP) on front side,

  front-end handle its own state

- 2011: SproutCore ➡️ Ember

The Front-end revolution

- 2010: Apple end support of Flash on iOS

- lodash creator bring Backbone.js

- Model–view–presenter (MVP) on front side,

  front-end handle its own state

- 2011: SproutCore ➡ Ember
- 2010: Angular JS

The Front-end revolution

- 2010: Apple end support of Flash on iOS

- lodash creator bring Backbone.js

- Model–view–presenter (MVP) on front side,

  front-end handle its own state

- 2011: SproutCore ➡️ Ember
- 2010: Angular JS
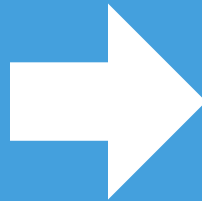- 2013: React

The Front-end revolution

- 2010: Apple end support of Flash on iOS

- lodash creator bring Backbone.js

- Model–view–presenter (MVP) on front side,

  front-end handle its own state

- 2011: SproutCore ➡️ Ember
- 2010: Angular JS
- 2013: React
- 2014: Angular X

The Front-end revolution

- 2010: Apple end support of Flash on iOS

- lodash creator bring Backbone.js

- Model–view–presenter (MVP) on front side,

  front-end handle its own state

- 2011: SproutCore ➡️ Ember
- 2010: Angular JS
- 2013: React
- 2014: Angular X

The Front-end revolution

- 2010: Apple end support of Flash on iOS

- lodash creator bring Backbone.js

- Model–view–presenter (MVP) on front side,

  front-end handle its own state

- 2011: SproutCore ➡️ Ember
- 2010: Angular JS
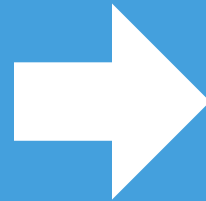- 2013: React
- 2014: Angular X

A JS dedicated Ecosystem

- package managers

- building tools

- libraries

Front-end revolution: Sum up

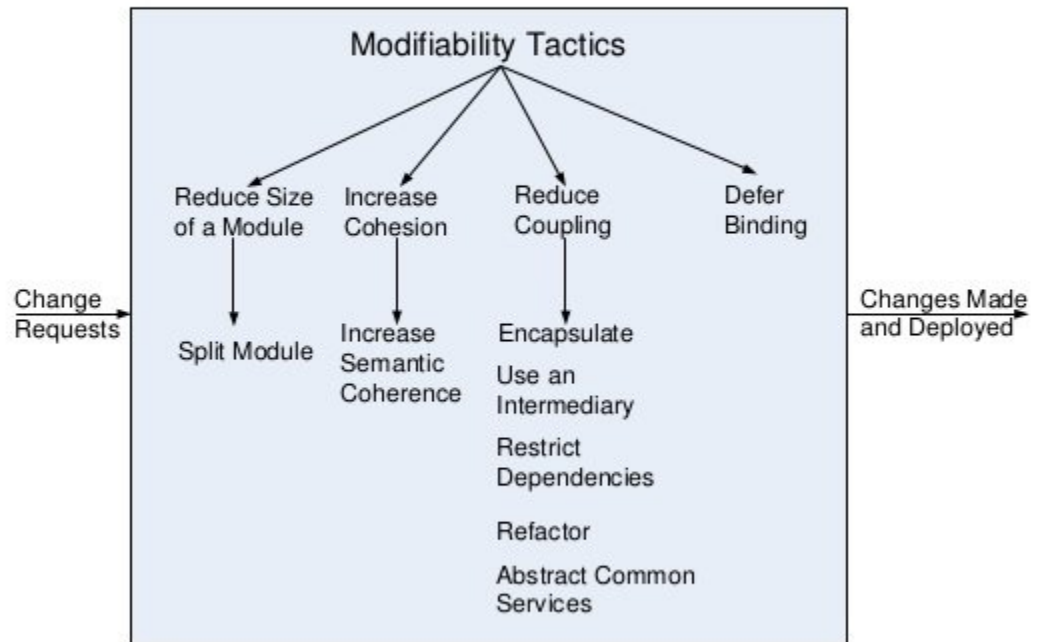| ① 00'	② 10'	③ 20'
JQuery / Prototype	**Backbone**	**Angular / React**
Cross-browsers API	MVP	MVV, MV*
Animations	Component State management	Separation of Concerns
-	Templating	Application State management
-	-	modules
		DI

How to do Architecture
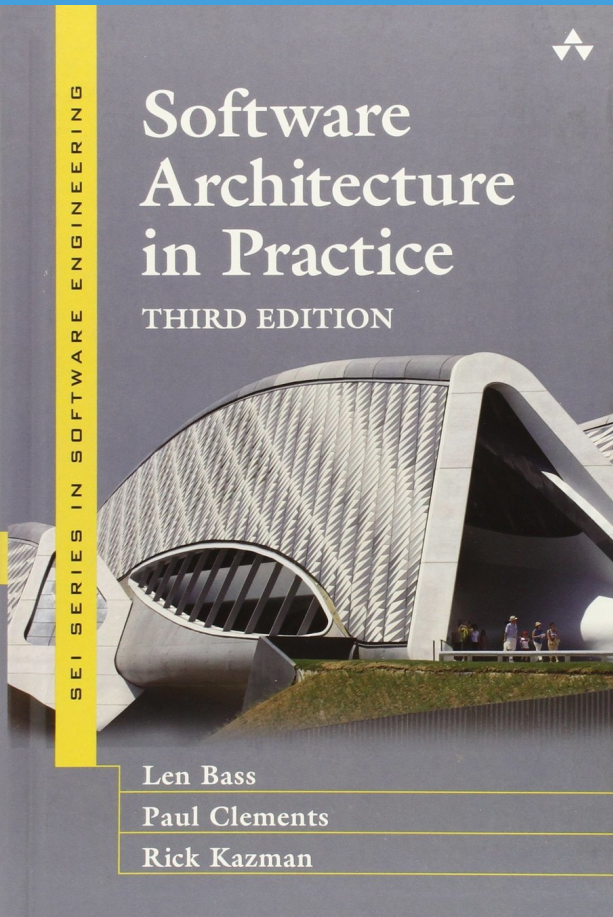
SEI SERIES IN SOFTWARE ENGINEERING

Software Architecture in Practice

THIRD EDITION

Len Bass
Paul Clements
Rick Kazman

How to do Architecture

Software Architecture in Practice
THIRD EDITION

SEI SERIES IN SOFTWARE ENGINEERING

Len Bass
Paul Clements
Rick Kazman

Modifiability Tactics

Modifiability Tactics

Reduce Size of a Module — Increase Cohesion — Reduce Coupling — Defer Binding

Change Requests →

Split Module

Increase Semantic Coherence

Encapsulate
Use an Intermediary
Restrict Dependencies
Refactor
Abstract Common Services
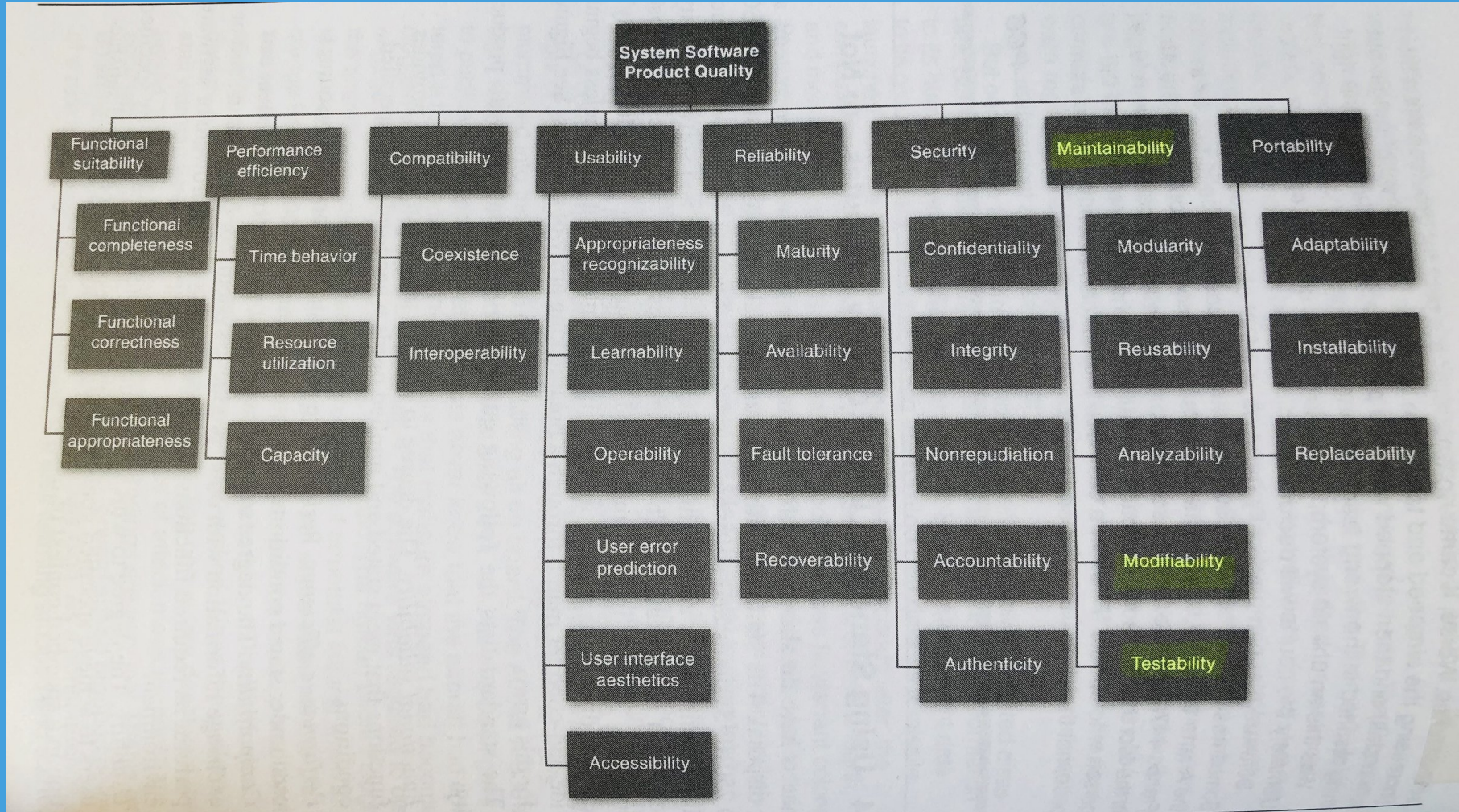
→ Changes Made and Deployed

What are Quality Attributes?

" *A Quality Attribute (QA) is a measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders*

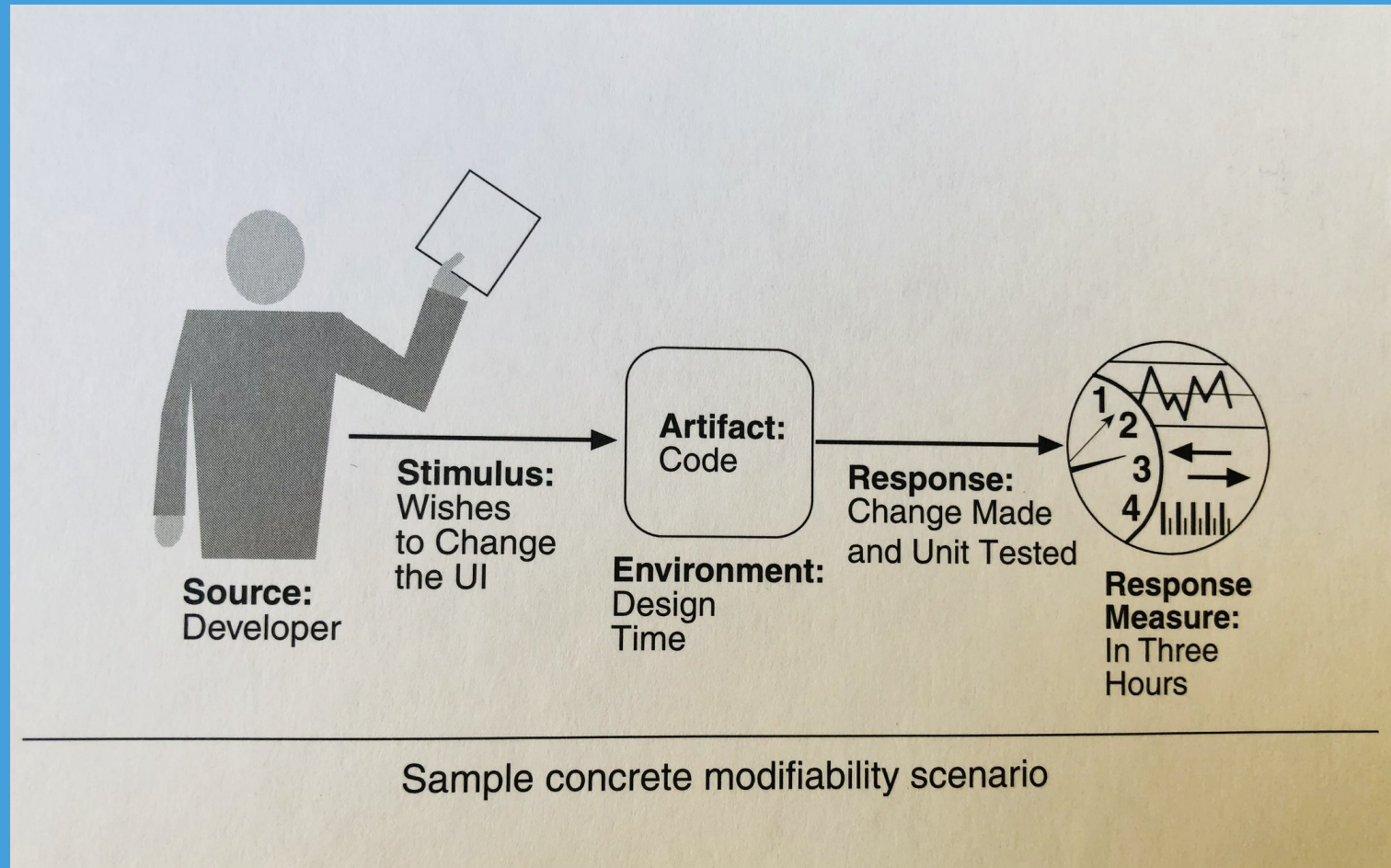*Software Architecture in Practice, p.63*

What are Quality Attributes?

System Software Product Quality tree diagram showing the following hierarchy:

- **Functional suitability**: Functional completeness, Functional correctness, Functional appropriateness
- **Performance efficiency**: Time behavior, Resource utilization, Capacity
- **Compatibility**: Coexistence, Interoperability
- **Usability**: Appropriateness recognizability, Learnability, Operability, User error prediction, User interface aesthetics, Accessibility
- **Reliability**: Maturity, Availability, Fault tolerance, Recoverability
- **Security**: Confidentiality, Integrity, Nonrepudiation, Accountability, Authenticity
- **Maintainability**: Modularity, Reusability, Analyzability, Modifiability, Testability
- **Portability**: Adaptability, Installability, Replaceability

Software Architecture in Practice, p.194

How Quality Attributes works?

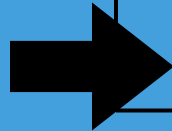Sample concrete modifiability scenario

Software Architecture in Practice

Apply QAs to front-end architecture

Product requirements + constraints

↓

Architecture QAs

→

Features updates

Technical roadmap

New features

Technical guide

A example of front-end architecture

- **Modifiability**
  Ability to introduce new features, refactor

- **Maintainability**
  Ability to onboard new developers, scale the code

- **Performance**
  User perceived performance

A example of front-end architecture

Build a technical roadmap

- TypeScript with types missing everywhere

  ❌ Modifiability, Maintainability ➡️ **type everything!**

- REST API, local custom redux cache

  ❌ Performance ➡️ **GraphQL migration**

- Components with bad naming

  ❌ Maintainability ➡️ **better naming**

- Complex generic class based components

  ❌ Modifiability, Maintainability ➡️ **refactor**

- Complex generic "models" with cache system

  ❌ Modifiability, Maintainability ➡️ **refactor**

A example of front-end architecture

Guidelines for new development

➡️ Enhance **Modifiability**, so :
reduce module size, increase cohesion and reduce coupling

- Components naming convention

- Redux as event-bus to defer binding of components

- Components refactoring to follow SRP principles

- Data HoC in order to prepare GraphQL migration

A example of front-end architecture

Guidelines for new development

Components naming convention

Domain, Page/Context, Component, Type

ACommunityAddToShortList**Button**
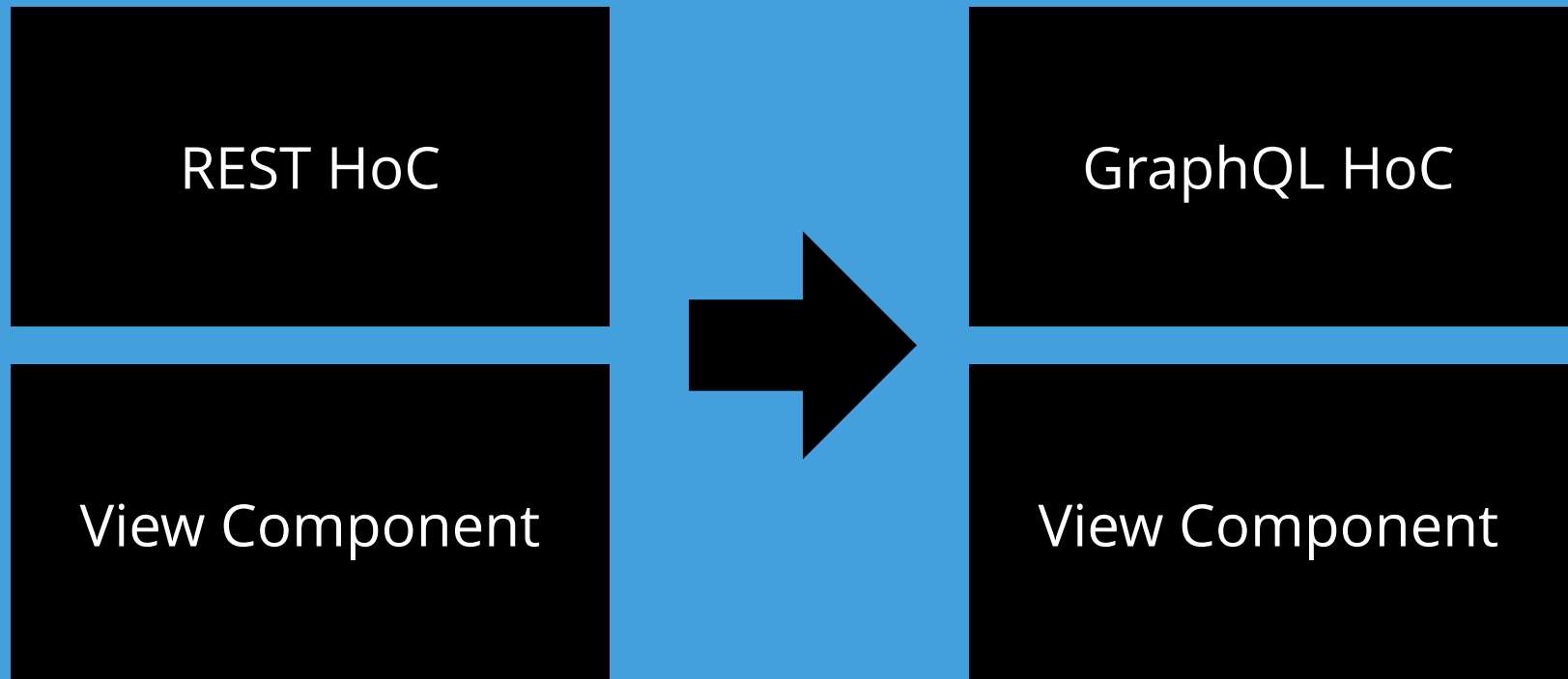
Sidebar

Sidebar**Switch**

Chat**Conversation**

Chat**ConversationName**

A example of front-end architecture

Guidelines for new development

The GraphQL migration anticipation

Conclusion

- We are here to solve problems

  - especially new problems on front side

- Think architecture, not code

- Architecture good points:

  - clear structure, easier to document

    - enhance team work, onboarding

  - early prediction of system's qualities

  - can be re-usable

Thanks for listening!

@whereischarly

/wittydeveloper

@wittydeveloper