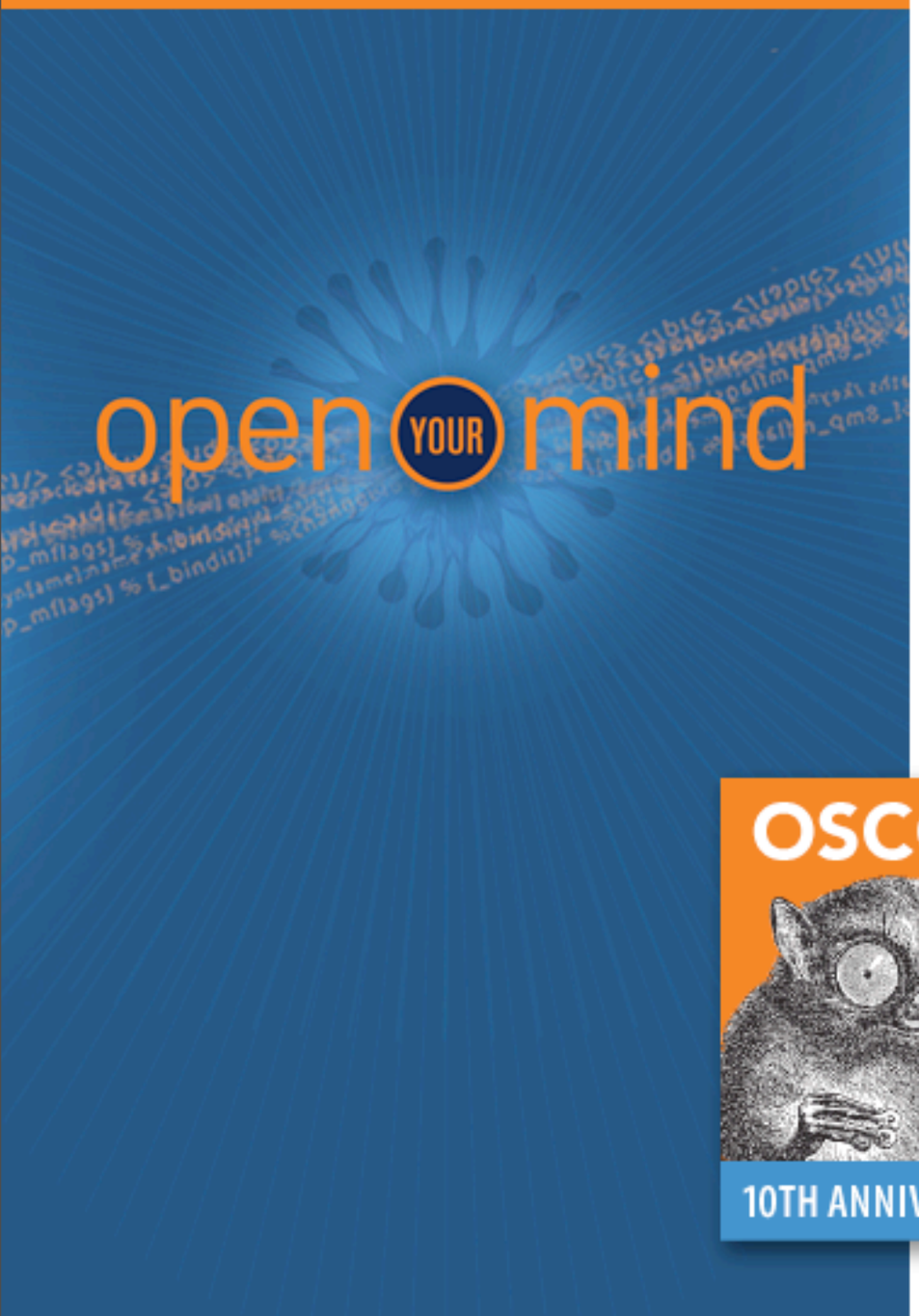
O'REILLY
OSCON
Open Source Convention



Open Architecture

Roy T. Fielding, Ph.D.
Chief Scientist, Day Software

http://roy.gbiv.com/talks/200807_OpenArch_OSCON.pdf

Open Architecture ?

Not talking about open exoskeleton buildings

Open Architecture ?

Not talking about
Open Sourced Architecture

Find Current Projects | Open Architecture Network

http://www.openarchitecturenetwork.org/projects

Google

Open Architecture Network beta 0.2

Register or Log in

Search

Welcome, guest: please register or log in to your account!

ABOUT | PROJECTS | COMPETITIONS | PEOPLE | RESOURCES

Find Current Projects

Start a Project

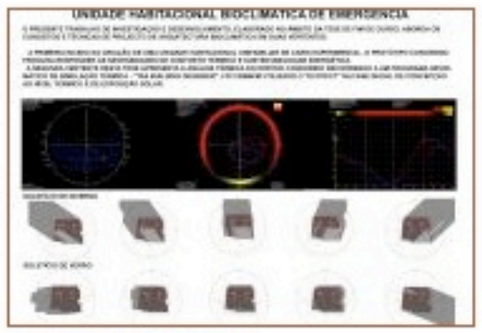
By Attributes

Search by keyword or use the Filter Criteria Search tools to narrow down your results.

1081 total projects.

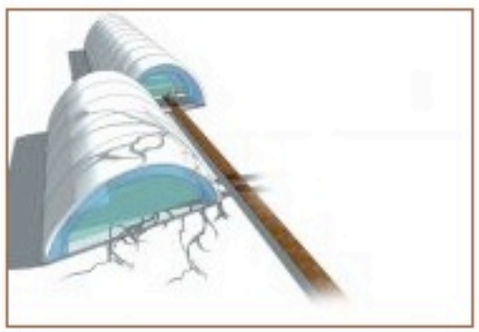
Sort by: Overall Rating

WatAir



Bioclimatic Shelter

Navajo Elder Straw Bale....



SEARCH PROJECTS

Search

Enter a keyword, i.e. "Cob" or "Straw"

FILTER CRITERIA SEARCH

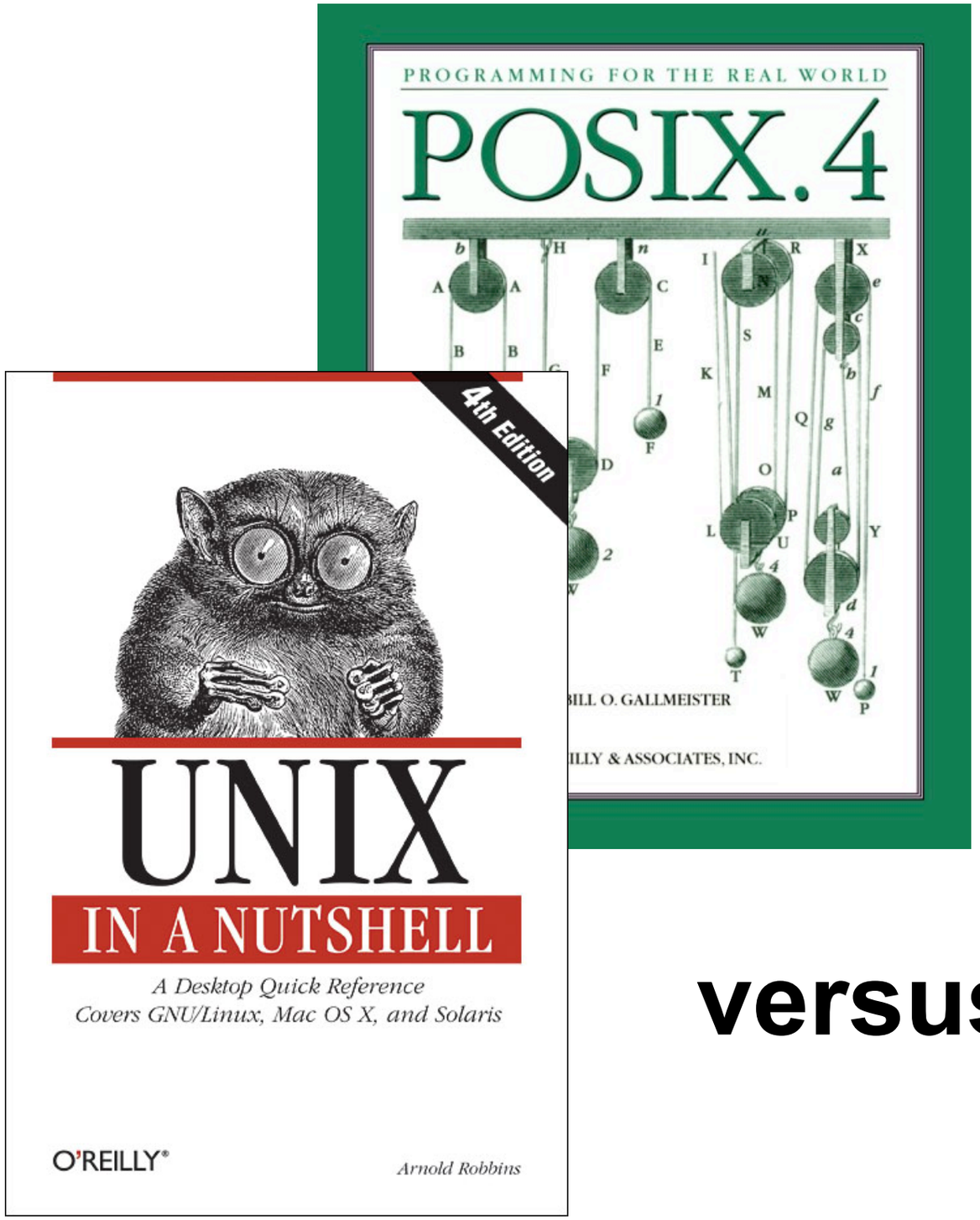
STATUS

All Statuses (1081)
Design development (1212)
Design complete (92)
In construction (54)
Construction complete (95)

REGION

All Regions (1081)
Africa & Arab States (190)
Asia & Pacific (175)
Europe & Former Soviet States (37)
Latin America & the Caribbean (110)

Open Architecture ?



Not talking about open systems

versus

Open Architecture ?

Not even talking about personal computer open architecture

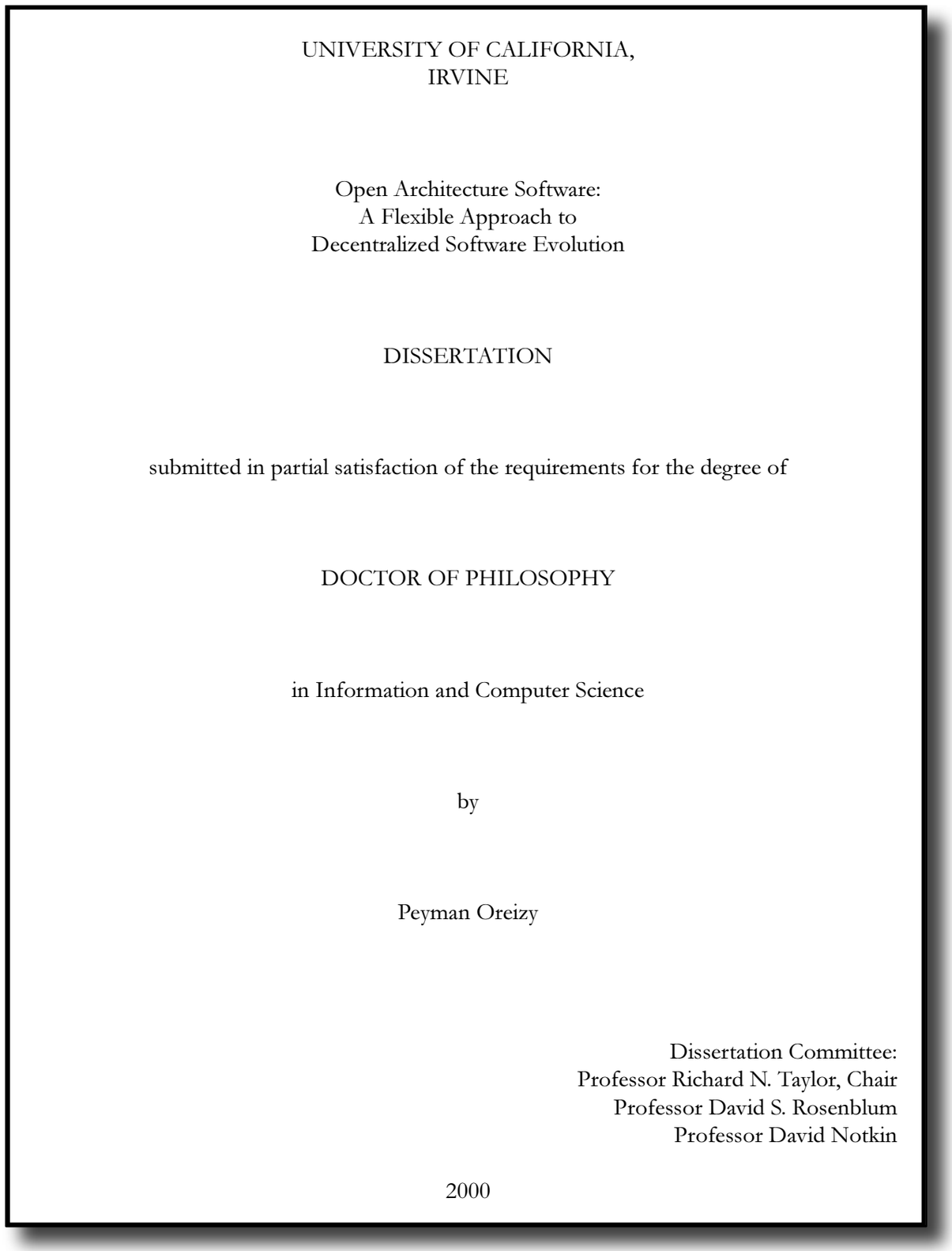
Open Architecture

Talking about another UCI Doctoral Dissertation!

Peyman Oreizy, 2000

Open Architecture Software: A Flexible Approach to Decentralized Software Evolution

http://www.ics.uci.edu/~peymano/



UNIVERSITY OF CALIFORNIA,
IRVINE

Open Architecture Software:
A Flexible Approach to
Decentralized Software Evolution

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Peyman Oreizy

Dissertation Committee:
Professor Richard N. Taylor, Chair
Professor David S. Rosenblum
Professor David Notkin

2000

Why talk about Open Architecture?

Open Development

Collaborative open source development

- emphasizes community
- takes advantage of the scalability obtainable through Internet-based virtual organizations
- adapts to the volunteer nature of developers

Why talk about Open Architecture?

Open Development

+

Conway's Law

Why talk about Open Architecture?

Open Development
+
Conway's Law

Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

Melvin E. Conway, *Datamation*, April 1968
http://www.melconway.com/law/index.html

Why talk about Open Architecture?

Open Development

+

Conway's Law

True open development
(a.k.a, Community-driven Design)
will only occur when the design of
your system reflects the organizational
structure of open development!

Why talk about Open Architecture?

Open Development

+

Conway's Law

+

Change is inevitable!

Why talk about Open Architecture?

Open Development

+

Conway's Law

+

Change is inevitable!

Why talk about Open Architecture?

Open Development

+

Conway's Law

+

Change is inevitable!

Decentralized Software Evolution
(or rapid obsolescence)

Decentralized Software Evolution

Requires Architecture by Design

open (software) architecture

used by open development projects

to enable decentralized software evolution

so that others can continue to design the software

and fulfill Conway’s Law

leading to success over time!

Techniques for Software Evolution

When

Who	Design-time (or pre-deployment) Evolution	Post-deployment Evolution
Central authority (e.g., single vendor)	Design notations, methods, and tools; process systems; group communication and collaboration; configuration management	Release management systems; binary patch files; configurable distributed systems
Decentralized group (e.g., multiple independent software vendors)	Same as above, with special support for loose coordination and collaboration; open source	APIs, software plug-ins, scripting languages, open source, component architectures, and event architectures [Oreizy 98, IWPSE 1]

Challenges

Trade-off between Adaptability and Consistency

> what changes are possible?

> what assurances are provided?

Where to place the open points

> behavioral junctions (APIs, callback hooks)

> virtual machines (command tables, scripting)

> data flow (filters, plug-ins)

Consistency

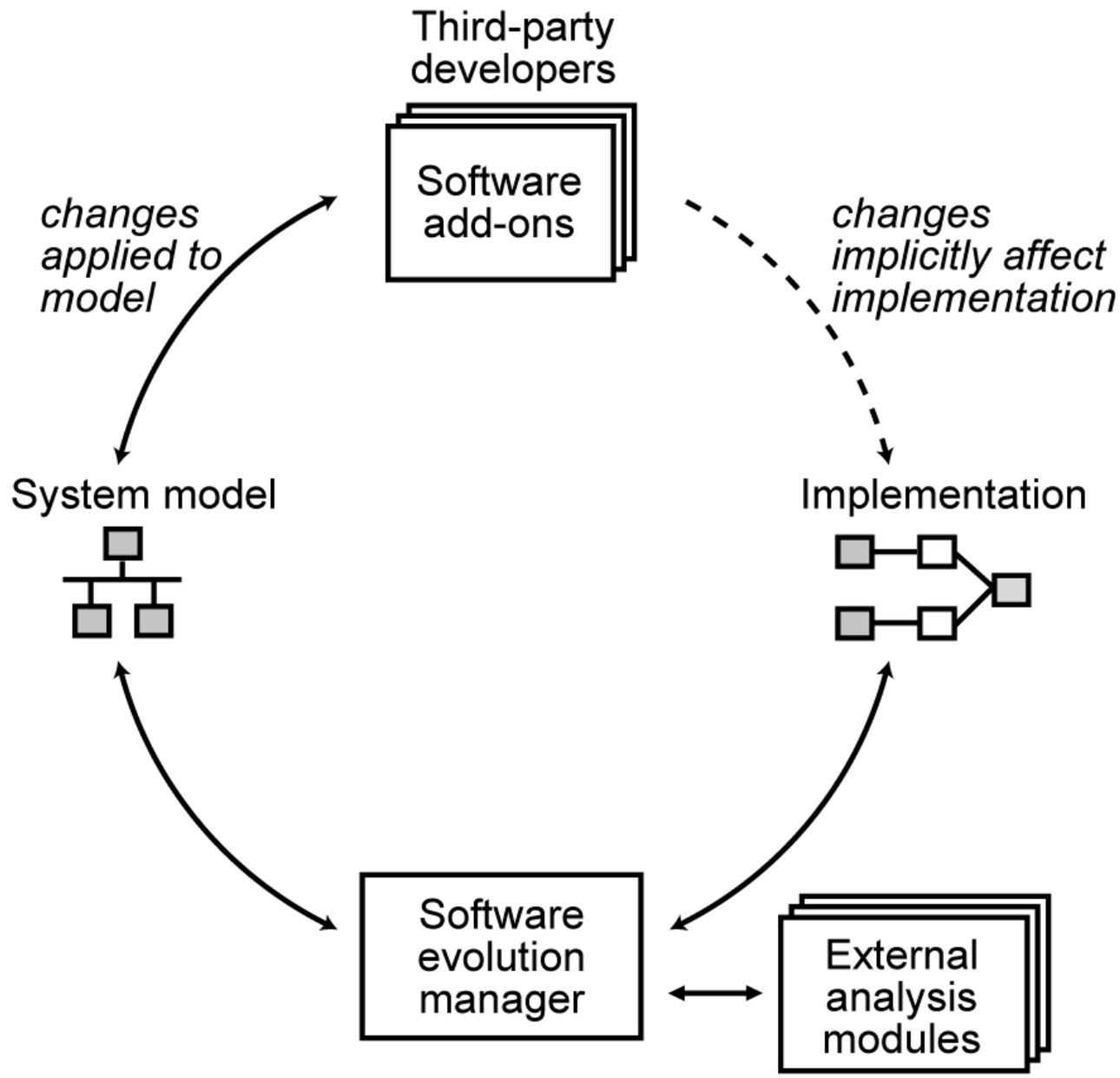
> independent third-parties may change the application in incompatible ways

> need for consistency may depend on type of system

– no data loss, response time, no feasting while others starve, no deadlocks, never too much radiation, …

Peyman's approach to Open Architecture

1. **Expose the application's architecture to third-parties**
2. **Allow third-parties to evolve the application by changing its architecture**
3. **Verify changes against the semantic annotations on the system model**
   - with the assistance of external analysis modules
   - if change is okay, make the corresponding change to the implementation
   - else, take appropriate action
     - notify, prevent, …

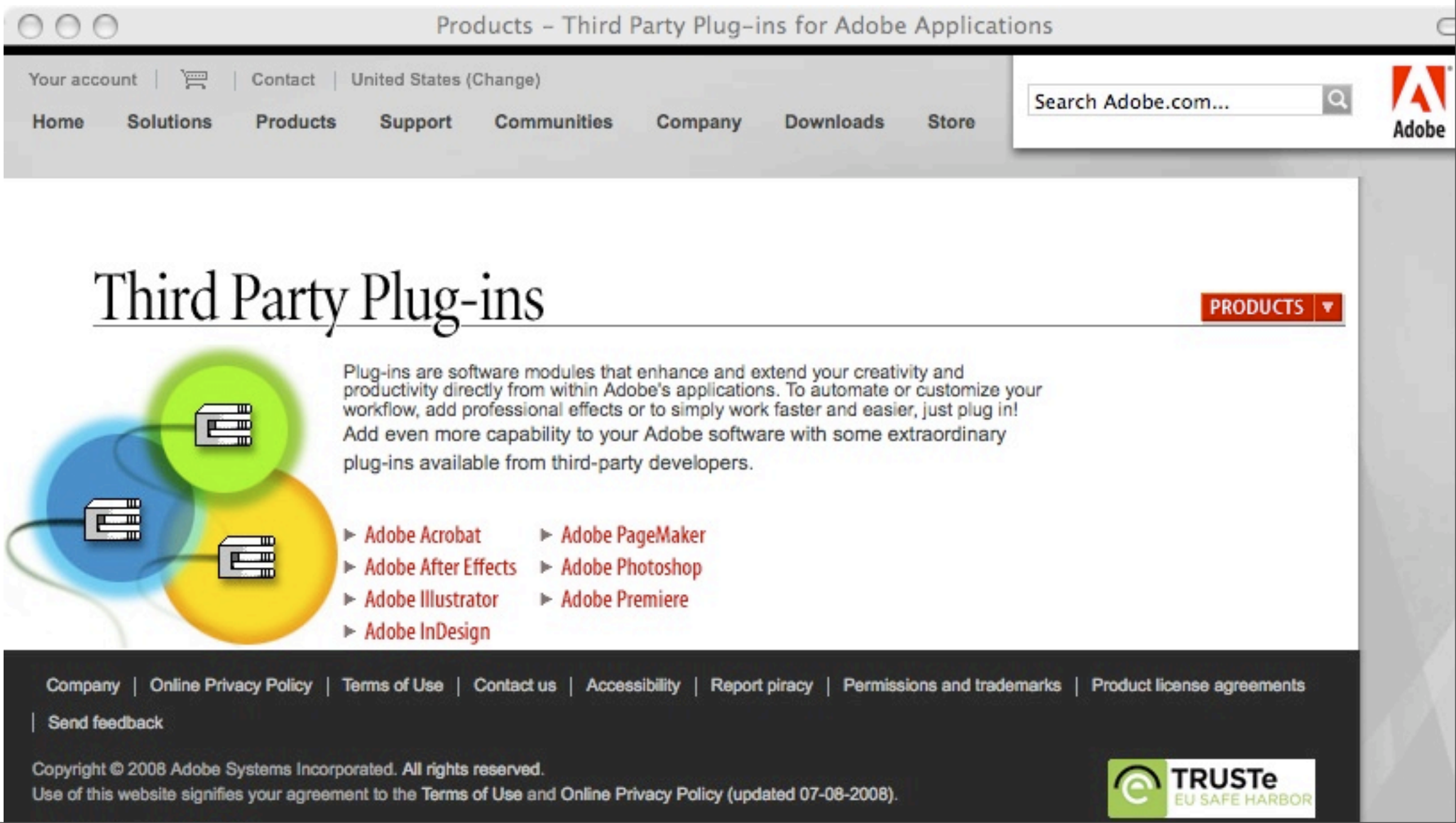


Closed Source Examples

Adobe

> extensive plug-in architecture

> diverse commercial community



Closed Source Examples

Apple iPhone Ecosystem

Open Development Software Evolution

What is common to all of the largest and most successful open source projects?

a software architecture

designed to promote anarchic collaboration

through extensions

while preserving control over the core interfaces

Open Source Examples

Emacs [Stallman 84]
> scripting language (elisp) + plug-in (Lisp function hooks)

Apache HTTP Server
> module hooks + filters + pools (for consistency)

Linux
> kernel modules + abstract APIs

Mozilla Firefox
> plug-ins + extensions + themes + XUL + CSS

Eclipse
> Everything is an OSGi plug-in

Apache Sling
> RESTful interaction on JCR hierarchy with OSGi plugins

Apache httpd

Started with NCSA httpd 1.3

> Simple, easy to compile on many legacy platforms
> Limited extensibility via CGI

Improved security, features, and performance

> Virtual hosts
> Pre-forking (adaptive hunt-group) model

0.8: re-architected for extensibility (Shambhala)

> Modular API for features (hook and ladder design)
> Pools for memory allocation (robustness)

2.0: architecture enhanced for more extensibility

> Modular Process Model (pre-fork, multithreaded, win32)
> I/O filters and protocol modules

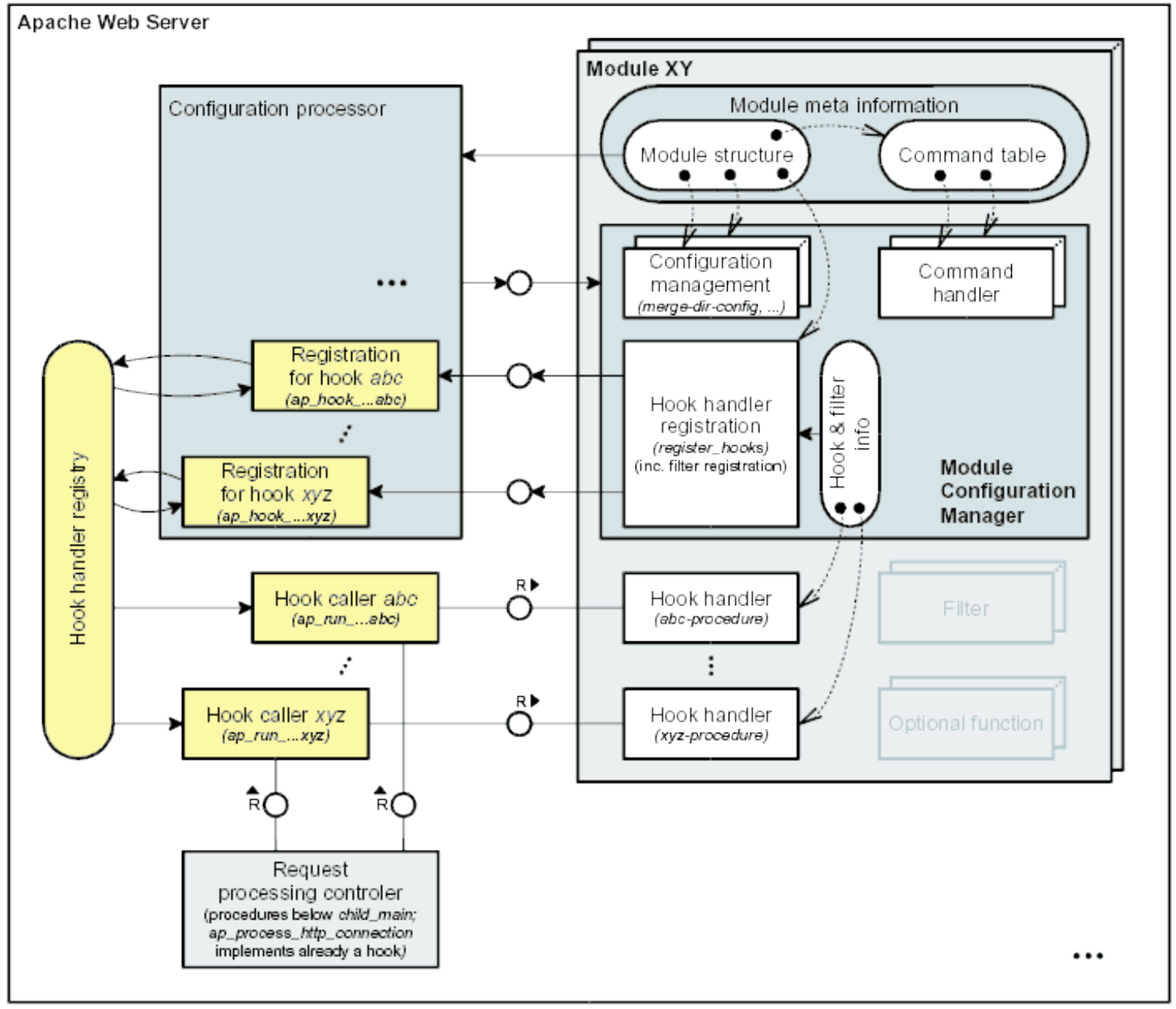
Apache httpd: modules

Modules

- **simplify core**
- **enable independent development**
- **promote experiments**

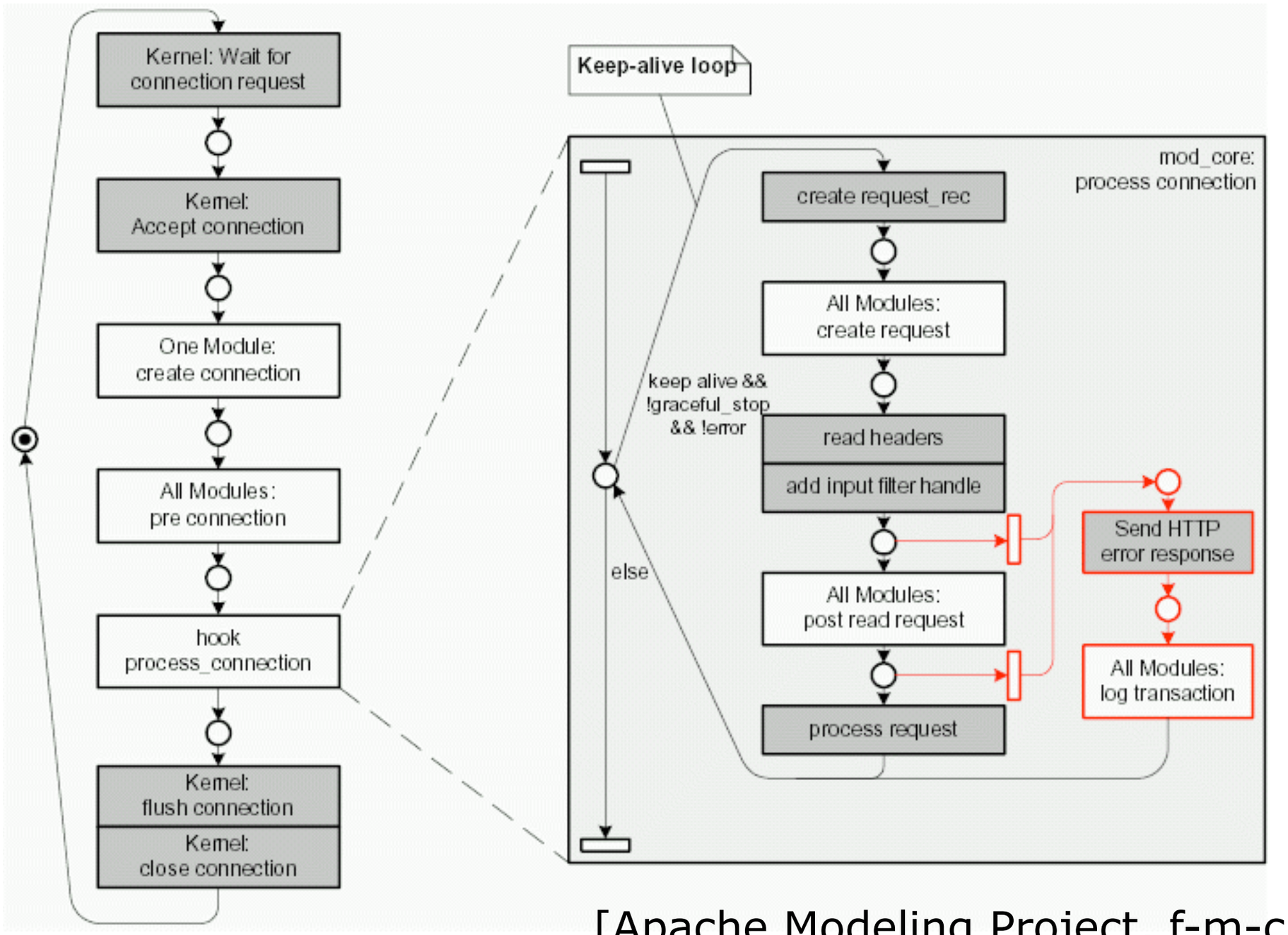
Project improves

- **reduced friction**
- **anarchic growth**
- **more features**
- **less communication**



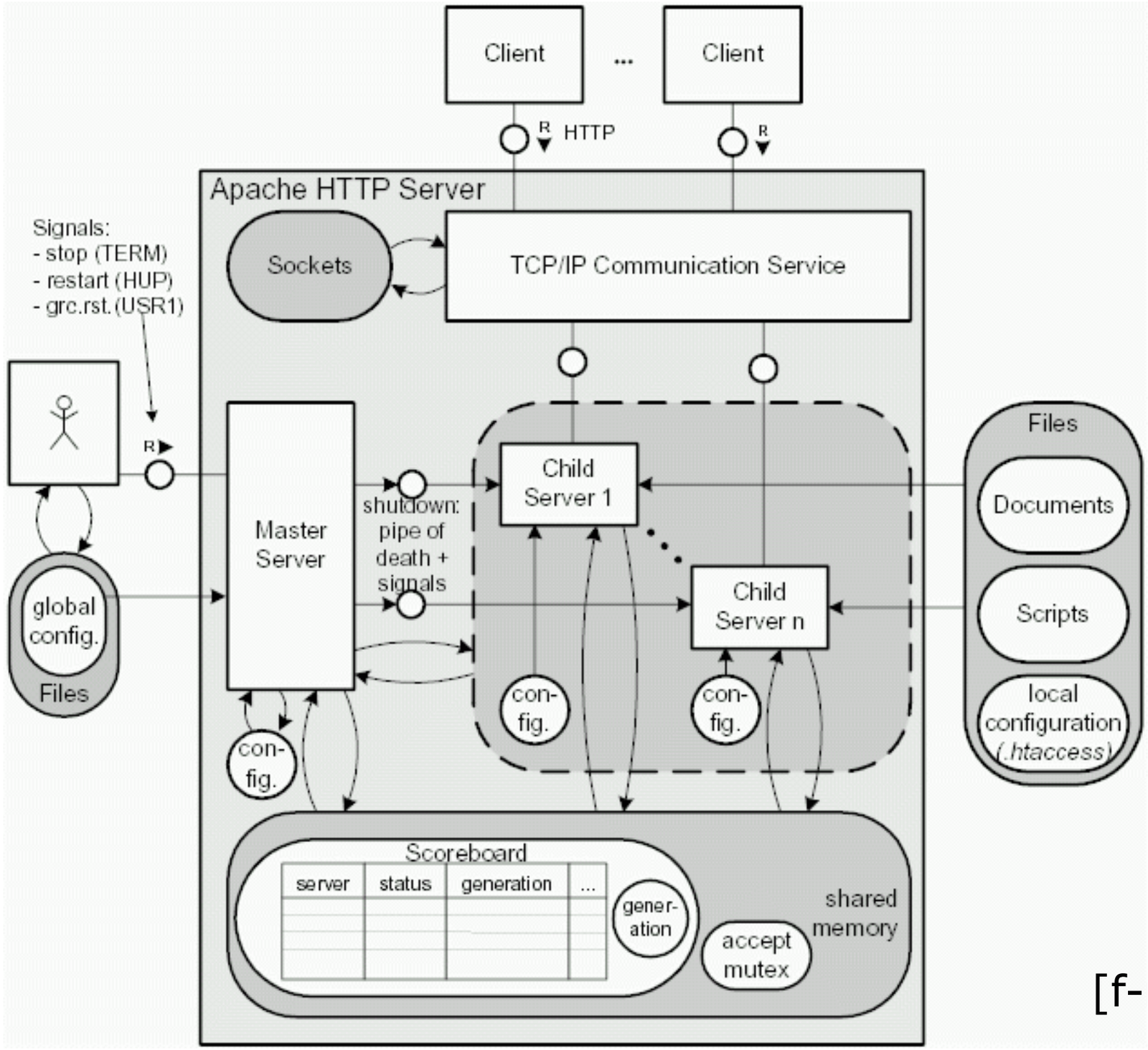
[Apache Modeling Project, f-m-c.org]

Apache httpd: kernel



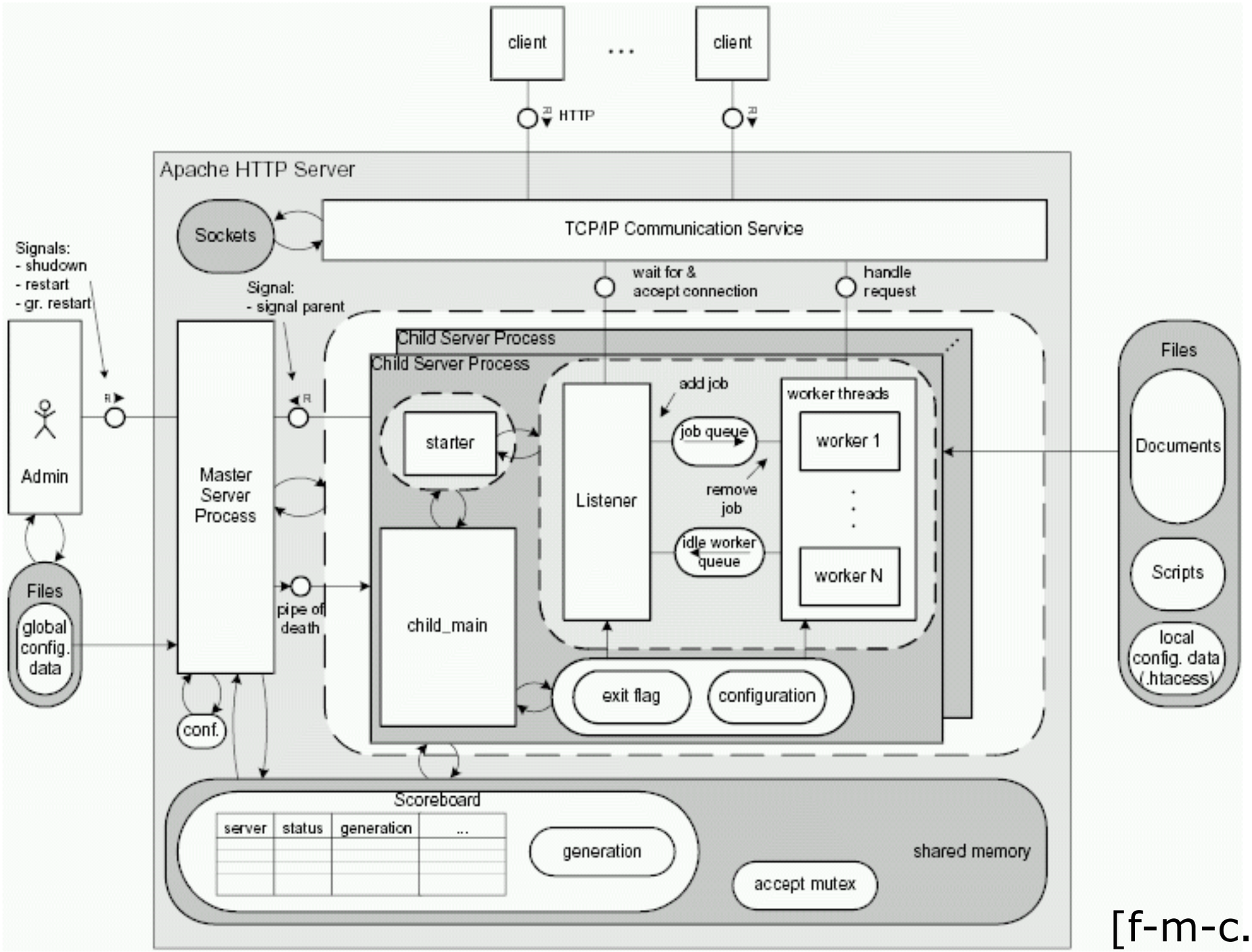
[Apache Modeling Project, f-m-c.org]

Apache httpd: preforking MPM



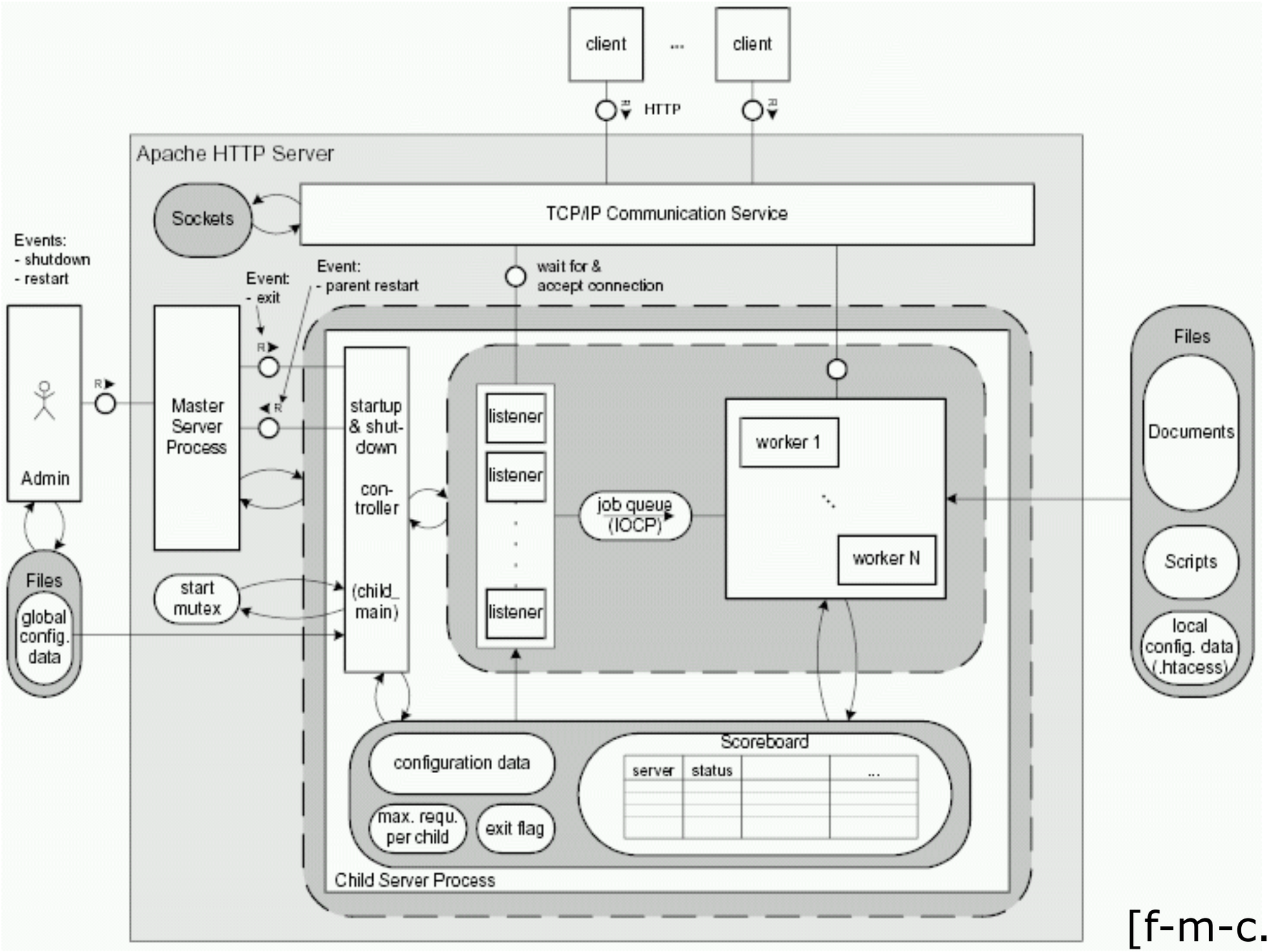
[f-m-c.org]

Apache httpd: worker MPM



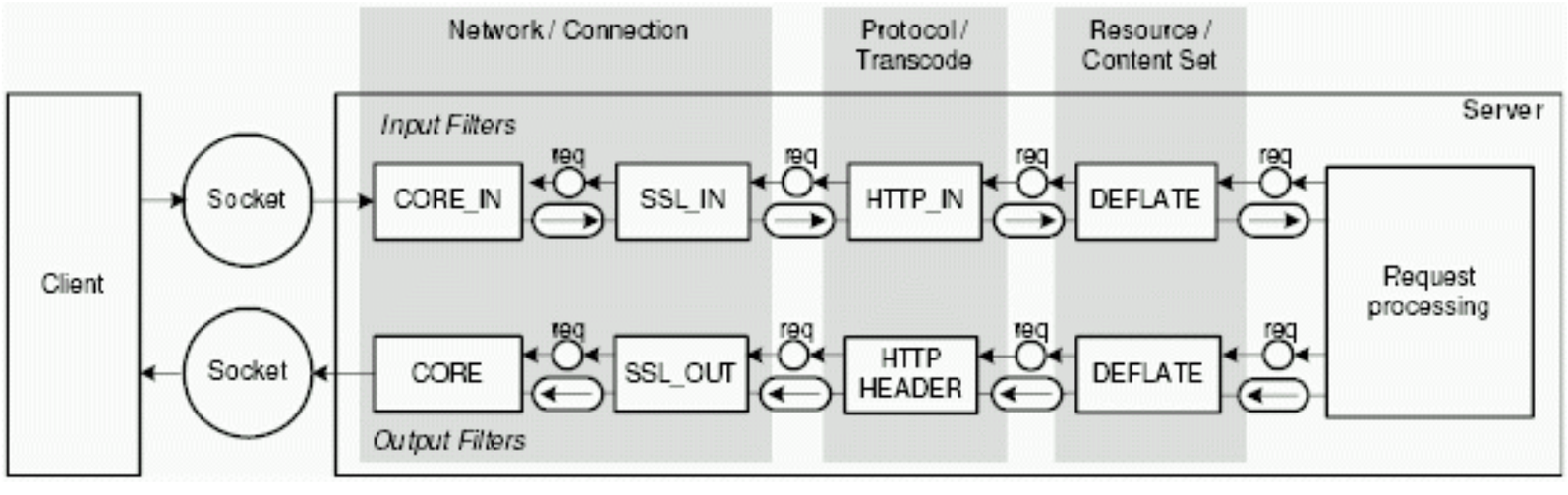
[f-m-c.org]

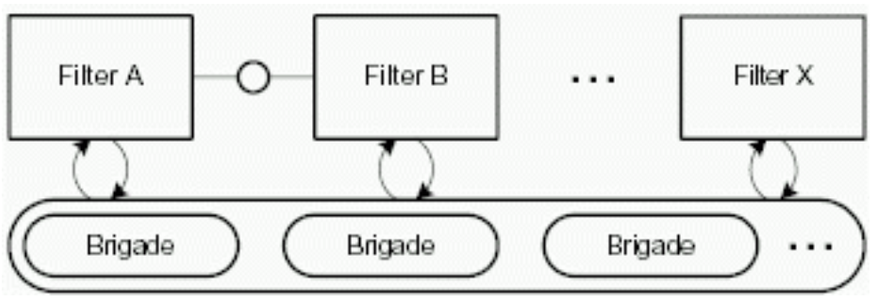
Apache httpd: winnt MPM



[f-m-c.org]

Apache httpd: I/O filters





Filters provide more extensibility

- **protocol replacement**
  - **httpd, ftpd, nntpd, ...**
- **stackable content manipulation**
  - **extensions that can extend other extensions**

[Apache Modeling Project, f-m-c.org]

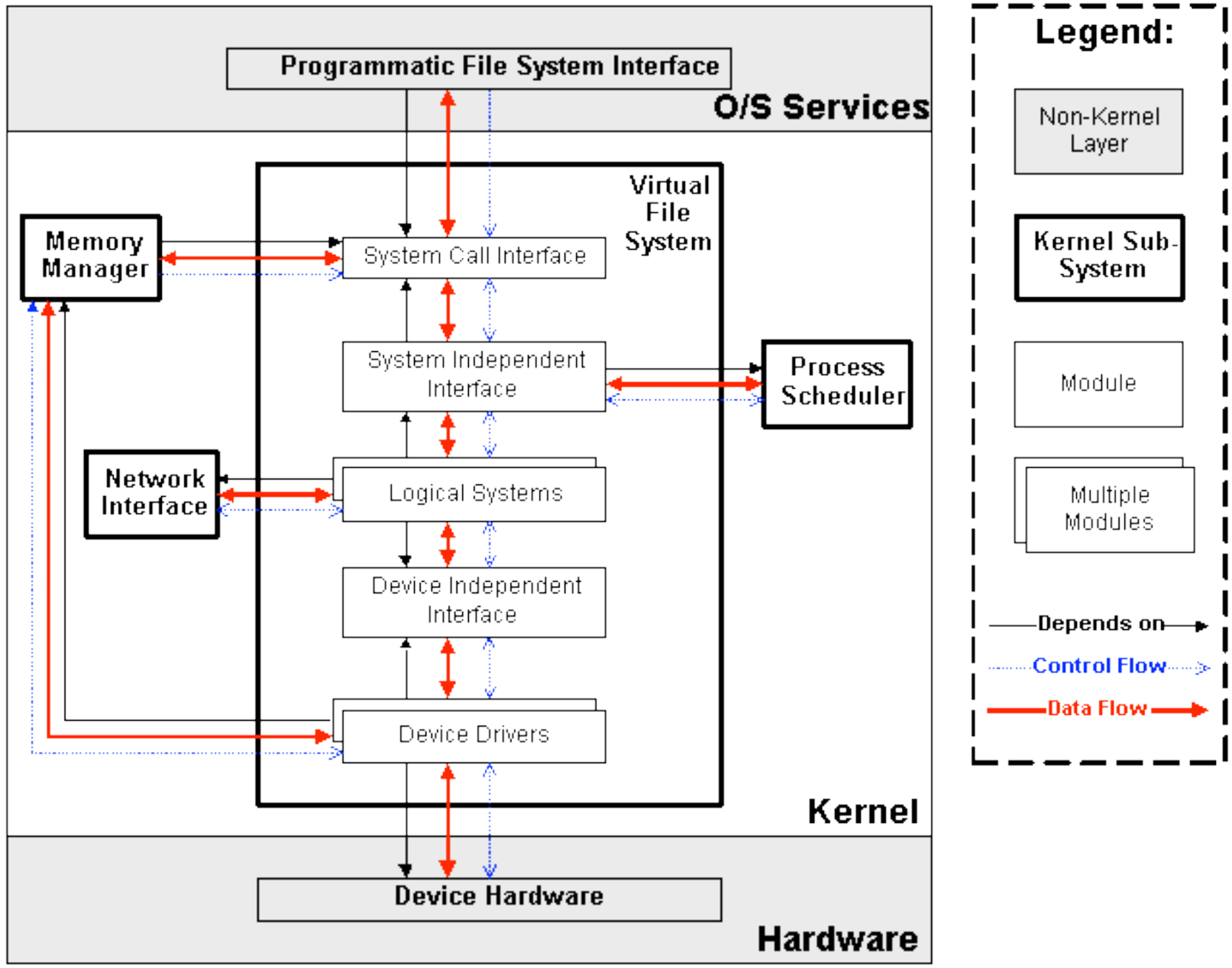
Linux Kernel Modules

Modules

- **simplify core**
- **enable independent development**
- **promote experiments**

Project improves

- **reduced friction**
- **anarchic growth**
- **more features**
- **less communication**



[diagram from Ivan T. Bowman, 1998]

Mozilla Firefox

Multiplatform

Standards Compliant

Lightweight

Community Supported

Firefox: User-friendly

Tabbed Browsing

Integrated Search

Live Bookmarks

RSS/XML Feeds

UI Themes

Firefox: Developer-friendly

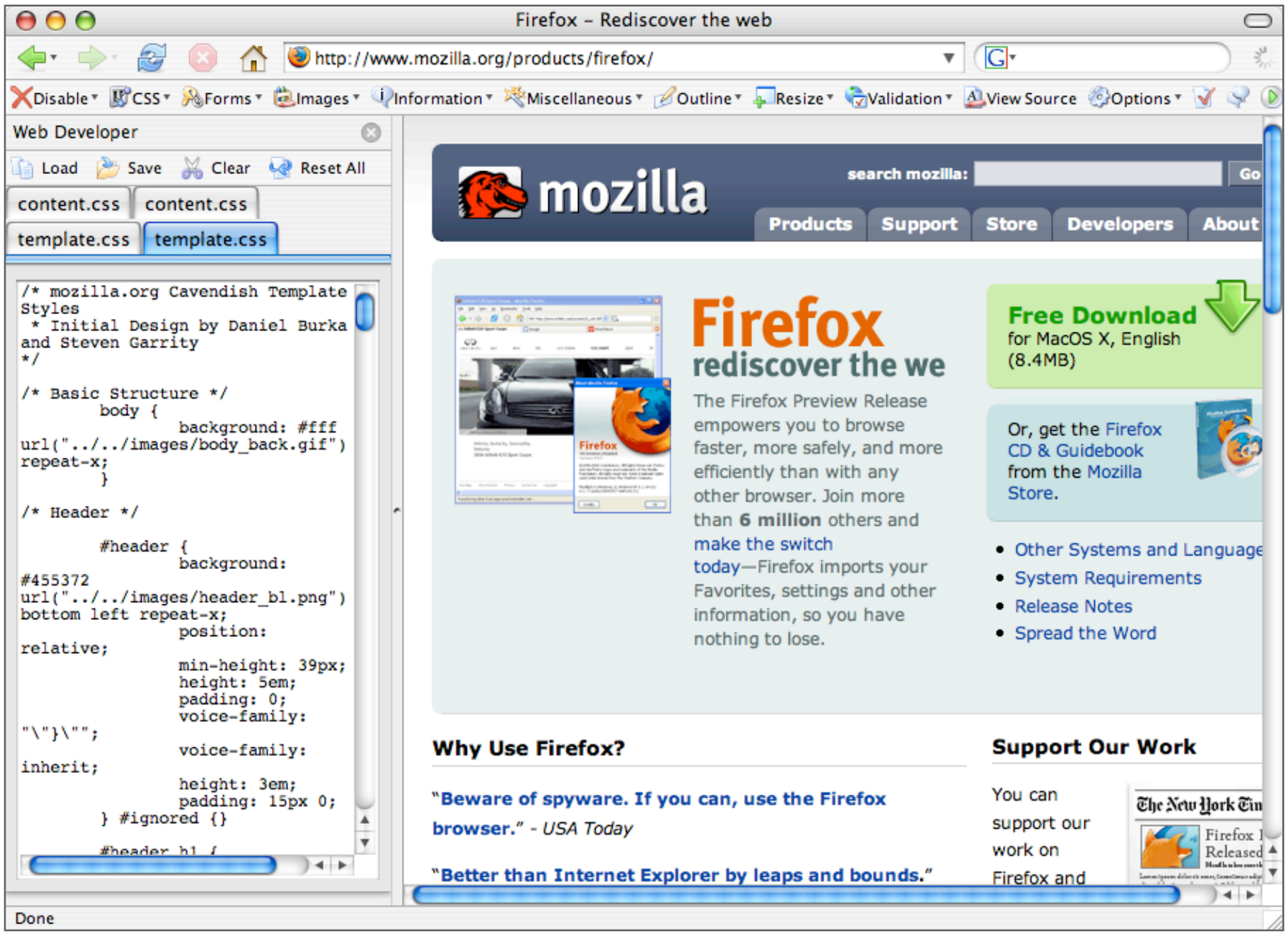
Open Source

Extensible Architecture

Plug-in Tools

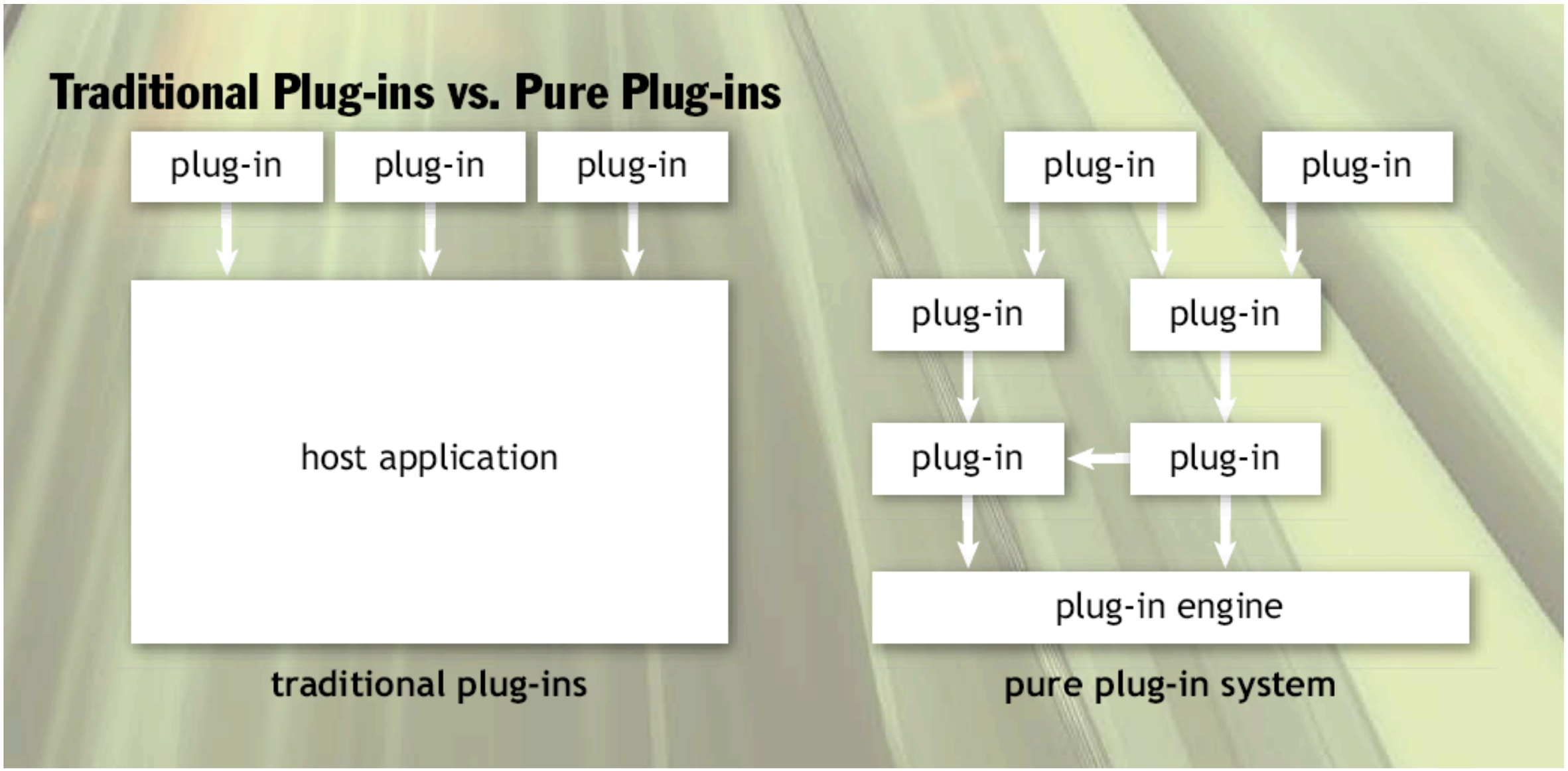
Layered CSS

Editor Platform



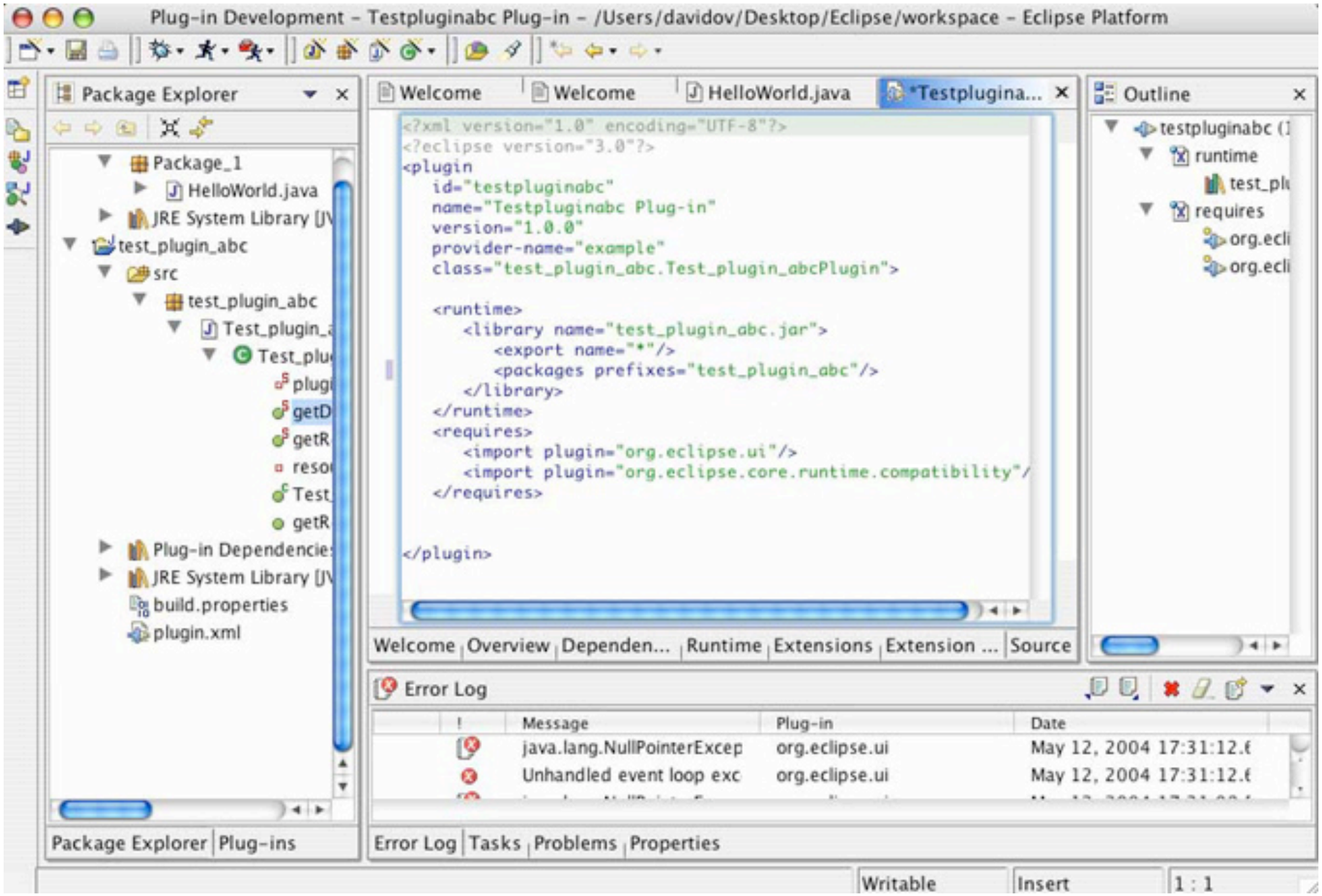
Eclipse Platform

Taking modular extensibility to the next level

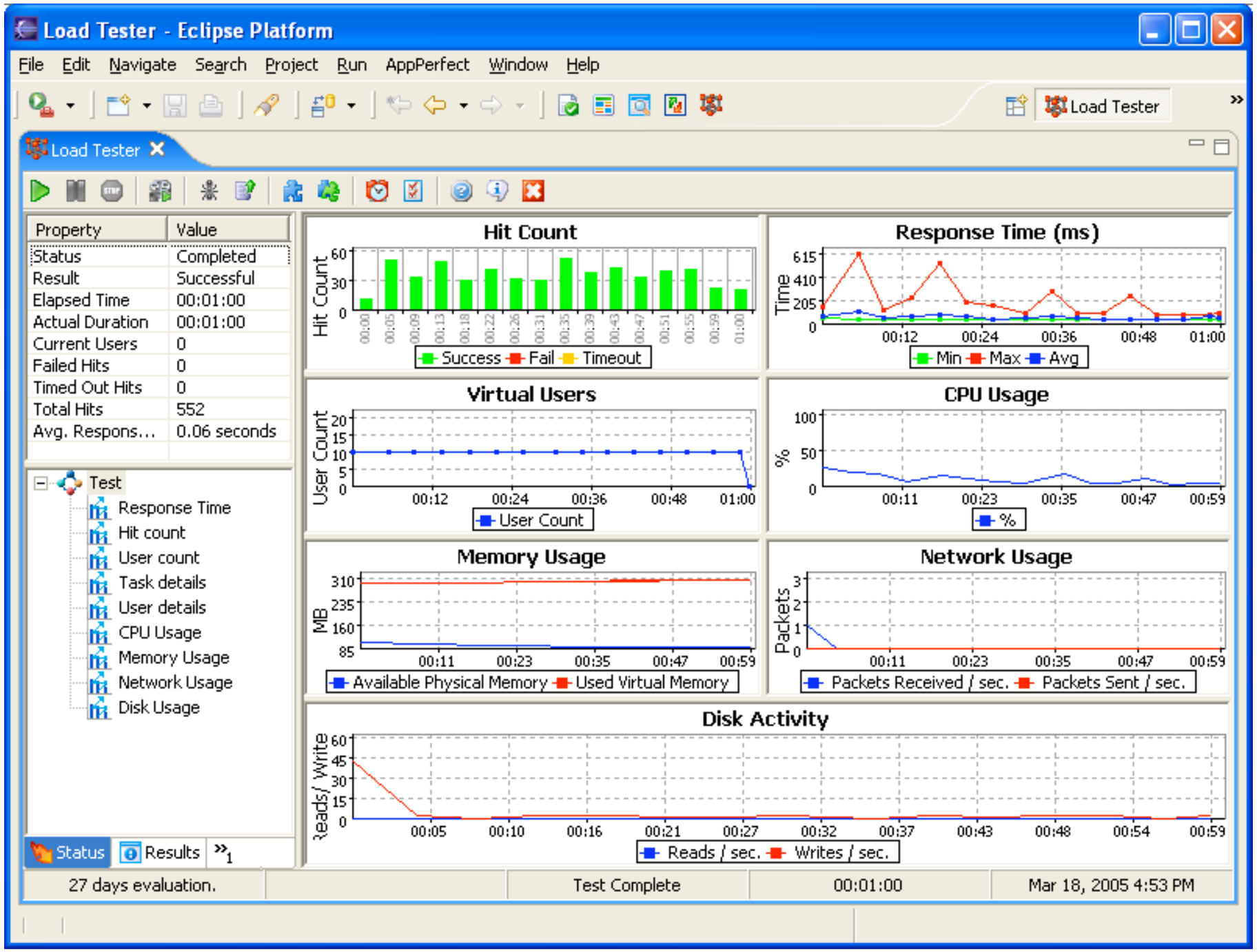


[Birsan, *ACM Queue*, Mar 2005]

Eclipse Platform



Eclipse Platform



Apache Sling

Drop-in Extensibility using OSGi Bundles

\+ JCR backed Content-oriented

WebDAV-able REST-based

Apache Sling

Apache incubator

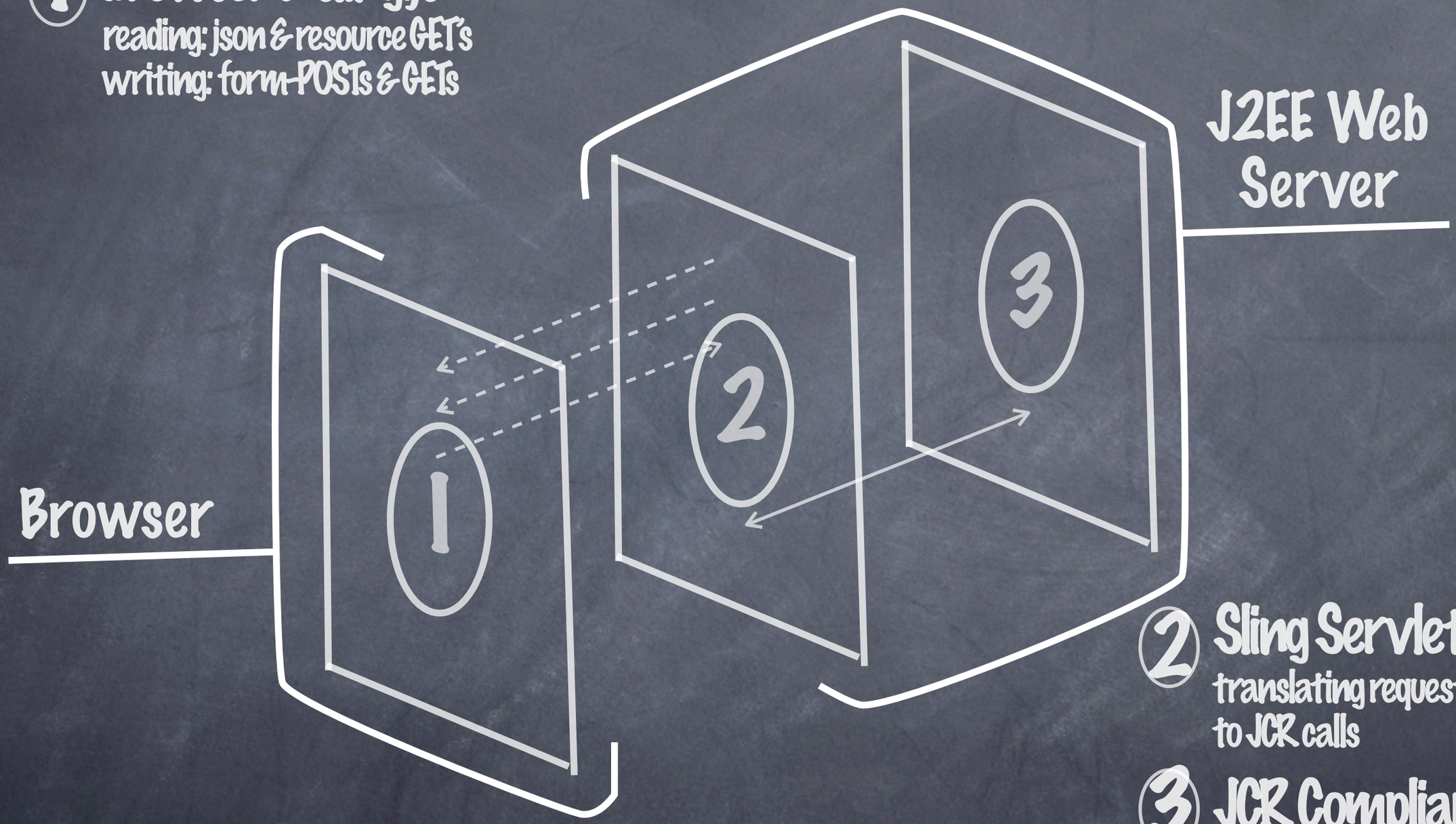
Drop-in Extensibility using OSGi Bundles

... jsp, groovy, esp, ruby, scala

+

JCR backed Content-oriented WebDAV-able REST-based

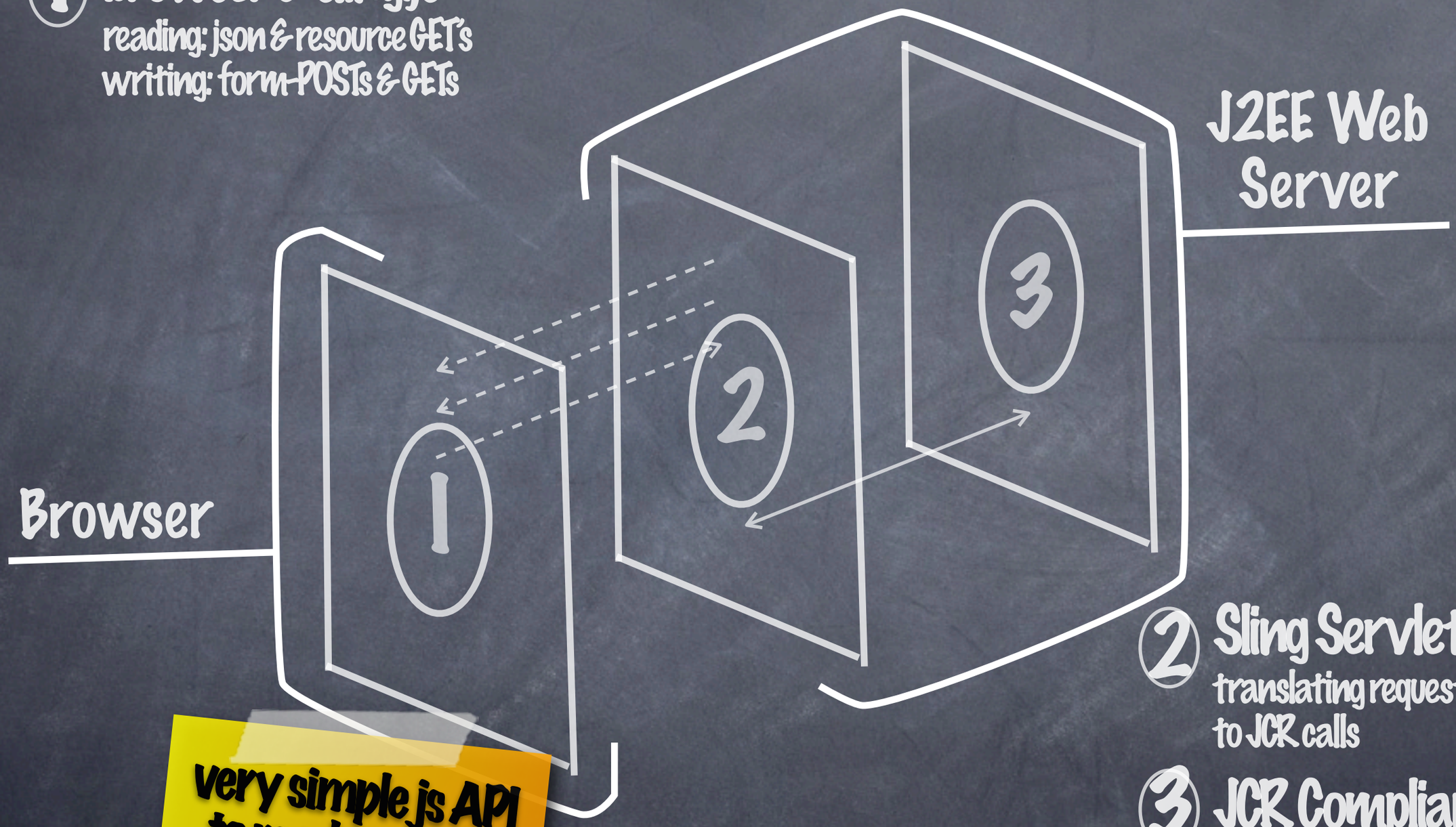
1 browser & sling.js
reading: json & resource GET's
writing: form-POSTs & GETs



2 Sling Servlets
translating requests to JCR calls

3 JCR Compliant Content Repository

1 browser & sling.js
reading: json & resource GET's
writing: form-POSTs & GETs

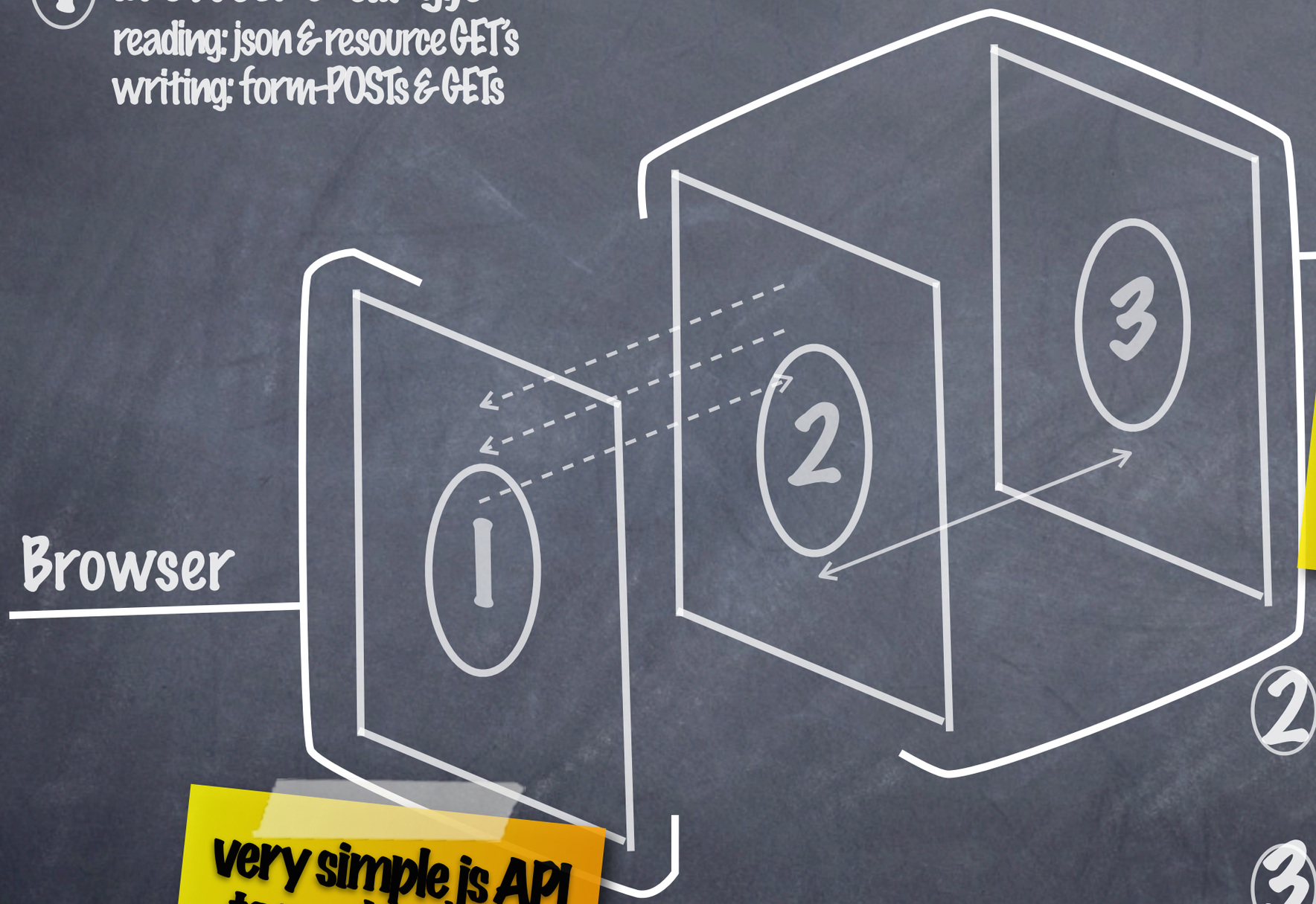


very simple js API to read content, Forms to write content

2 Sling Servlets
translating requests to JCR calls

3 JCR Compliant Content Repository

1 browser & sling.js
reading: json & resource GET's
writing: form-POSTs & GETs



J2EE Web Server

handles all the heavy lifting, particularly security

very simple js API to read content, Forms to write content

2 Sling Servlets
translating requests to JCR calls

3 JCR Compliant Content Repository

It's the web!
URLs matter.

```
.../product.jsp?id=12346
```

It's the web!
URLs matter.

```
.../product.jsp?id=12346
```

Mistake 1:
Addressing the "Script"

It's the web!
URLs matter.

.../product.jsp?id=12346

Mistake 1:
Addressing the
"Script"

Mistake 2:
.jsp? What the heck?

It's the web!
URLs matter.

.../product.jsp?id=12346

Mistake 1:
Addressing the
"Script"

Mistake 2:
.jsp? What the heck?

Mistake 3:
Passing in "this"

Day

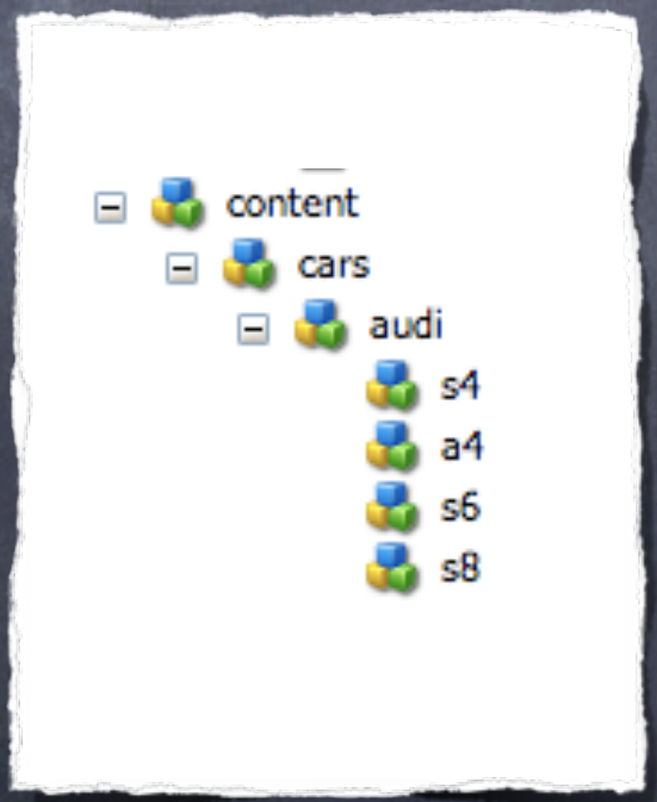
Content-oriented Development
Sling URL decomposition.

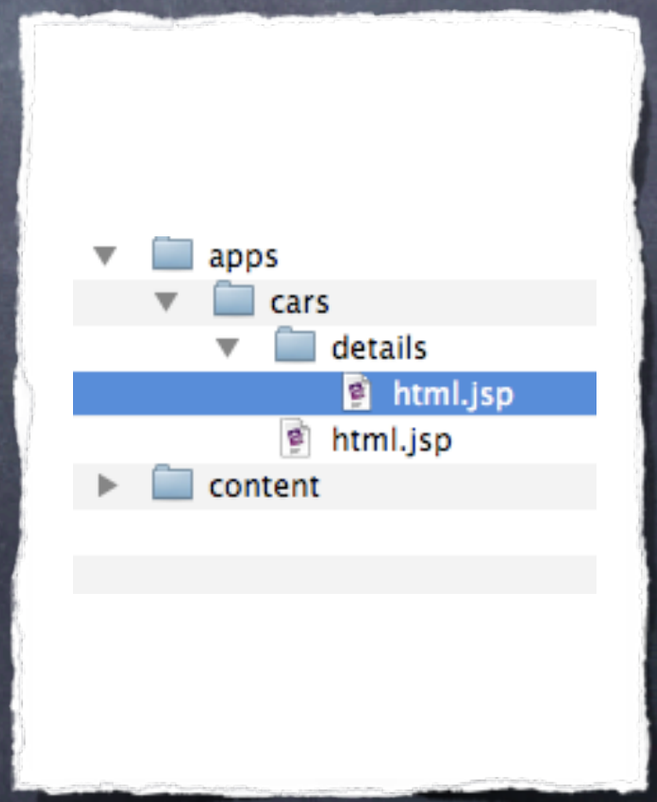
```
/cars/audi/s4.details.html
```

Content-oriented Development
Sling URL decomposition.

/cars/audi/s4.details.html

Repository



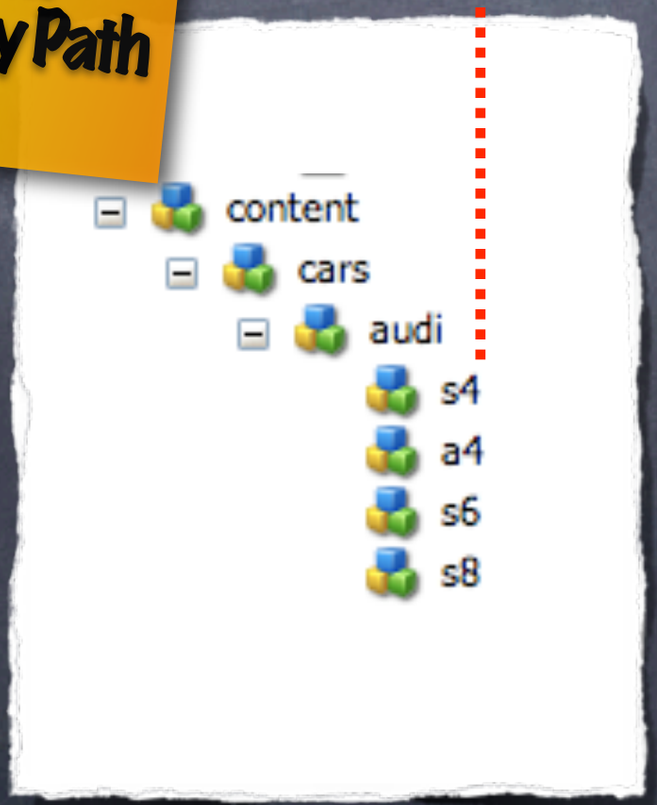


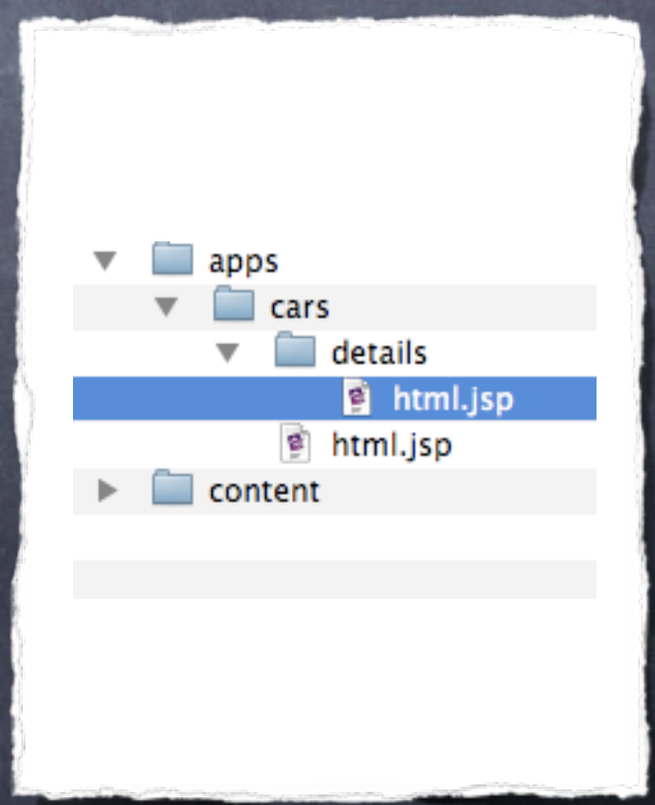
Content-oriented Development
Sling URL decomposition.

```
/cars/audi/s4.details.html
```

Content Repository Path

Repositor



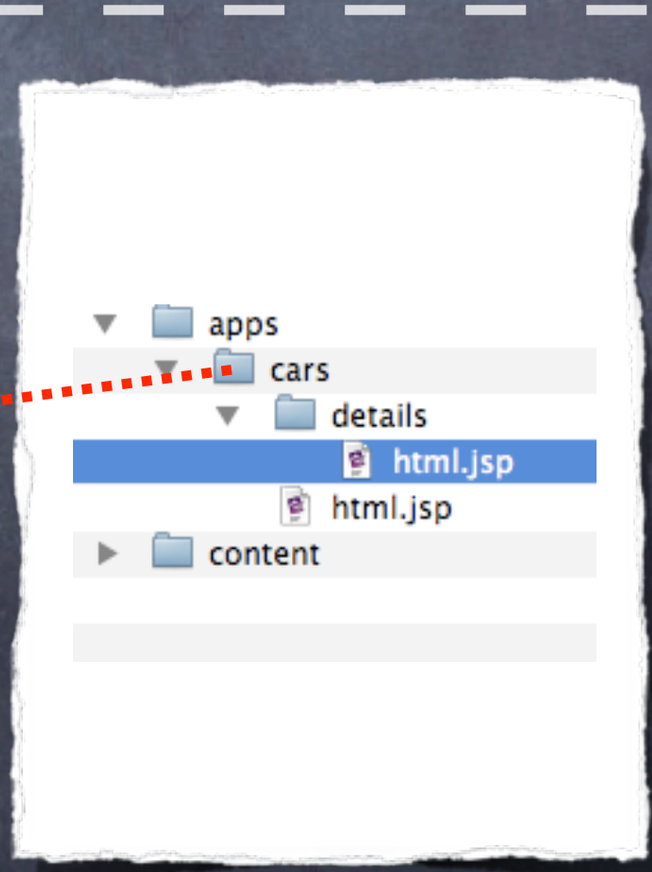


Content-oriented Development
Sling URL decomposition.

```
/cars/audi/s4.details.html
```

Repositor

Content Repository Path



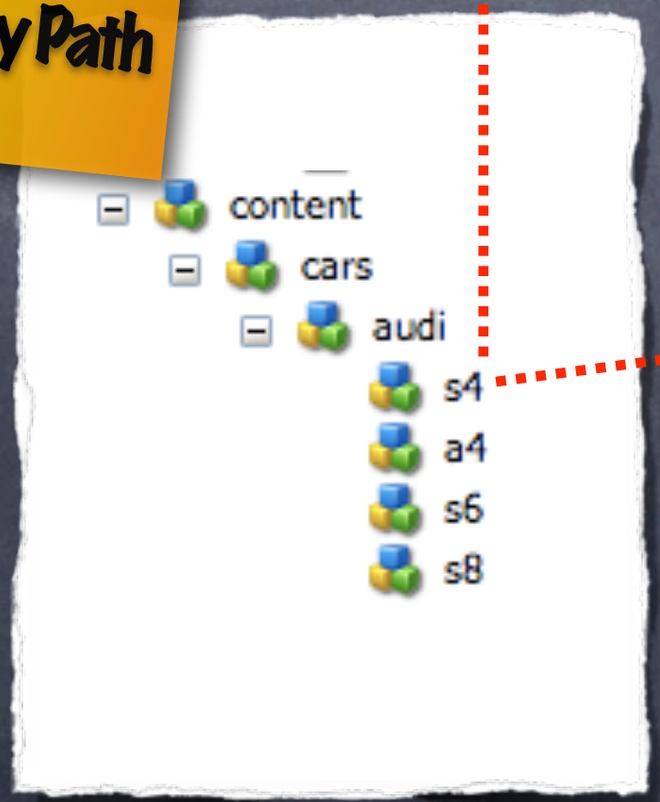
Content-oriented Development
Sling URL decomposition.

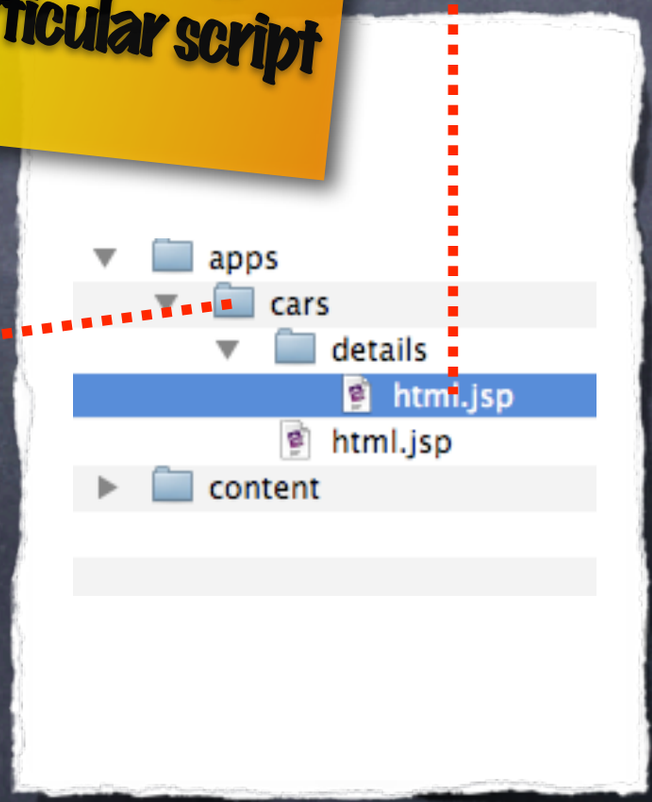
/cars/audi/s4.details.html

Content Repository Path

...selects a particular script

Repository





Conclusions

The best open source projects have learned

> the importance of designing an open architecture

> the value of decentralized software evolution

Or, perhaps we should say

> that's what differentiates the best from the obsolete!

There is so much more to learn

> I hardly scratched the surface of Peyman's dissertation

> many more successful examples

> how do we compare one design for over another?

> is OSGi bundling significantly better than others (for Java)

And I ignored more fundamental paradigm changes

> Erlang (designed around a shared-nothing architecture)