Striving for Versatility in Publish/Subscribe Infrastructures

Roberto S. Silva Filho and David F. Redmiles
{rsilvafi, redmiles}@ics.uci.edu

Department of Informatics
Donald Bren School of Information and Computer Sciences
University of California, Irvine

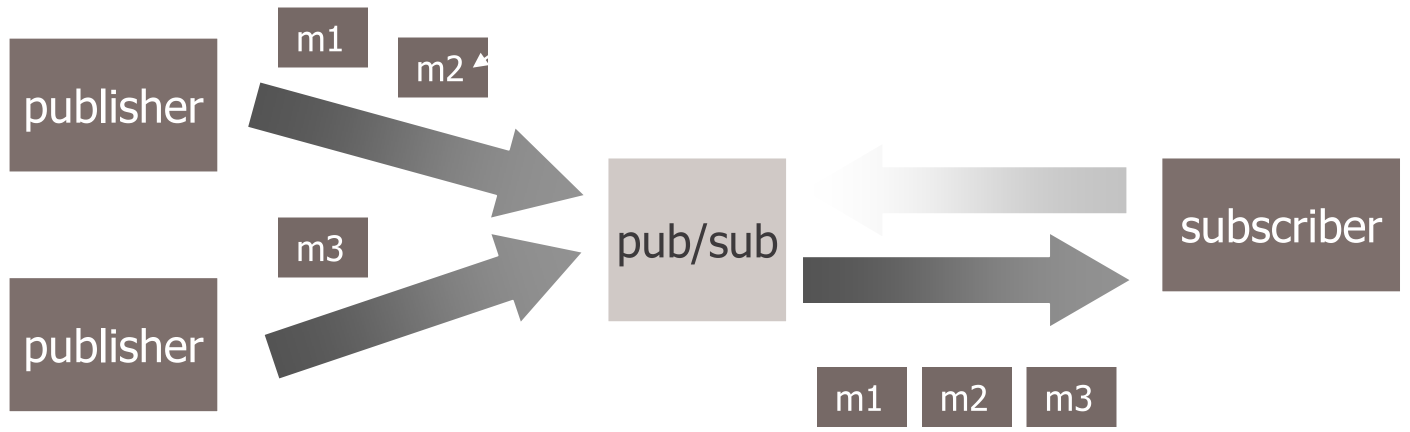
Presented at SEM 2005 – September 4th, 2005

Outline

- Motivation
- Versatility definition
- Approach
- Architecture overview
- Case studies
- Conclusions and future work

Motivation

- The publish/subscribe communication style provides:
  - data, flow and timing decoupling between producers and consumers of information
  - content-based filtering and communication
- This mechanism is usually implemented by a logically centralized infrastructure
  - intermediates the communication between publishers and subscribers in a distributed setting.



Motivation (continuation)

- For such properties publish/subscribe middleware has been used in different application domains such as:
  - software monitoring, groupware, workflow management systems, software development and deployment, mobile applications and so on.
- This wide range of applications have required different sets of services from the publish/subscribe infrastructure such as:
  - Advanced event processing, guaranteed event delivery, transactions, event persistency, secure communication channels, authentication, mobility support and many others

Motivation (continuation)

- In order to implement a distributed event-driven application, two main alternatives exist:
  - Use existing publish/subscribe infrastructures:
    - Standardized one-size-fits-all solutions: CORBA-NS or JMS
    - Minimal content-based routers such as ELVIN, SIENA, HERALD
  - Build new specialized pub/sub system
    - example: CASSIUS, GEM, YEAST and others.

Motivation (continuation)

- Those strategies, however, suffer from a fundamental problem:
  - They are not **flexible** enough [c.f. Parnas] :
    - They are usually not designed for change and evolution,
    - Nor to be expanded and contracted to address specific application needs

- Which results in:
  - The need for direct source code modification of existing solutions (when available)
  - The implementation of additional features at the application level
  - the build of new pub/sub infrastructures
    - resulting in the proliferation of incompatible proprietary infrastructures that are costly to evolve and maintain

Versatility

- In other words, current publish/subscribe infrastructures are not versatile enough to support their use in different application domains.

- Our concept of versatility comprises a set of properties:
  - Support for Evolution
    - Extensibility – add new functionality to the existing set
    - Programmability – redefine software behavior
    - Reuse
  - Support for Variability (footprint configuration)
    - Static (build or design time)
    - Dynamic (runtime)
  - Usability
    - Considerations about workplace environment
    - Nielsen’s attributes: learnability, efficiency, memorability, few errors and satisfaction.
  - Preserving middleware requirements of:
    - Scalability, interoperability, heterogeneity and communication

Our approach:

YANCEES, a versatile publish/subscribe infrastructure

Approach main characteristics

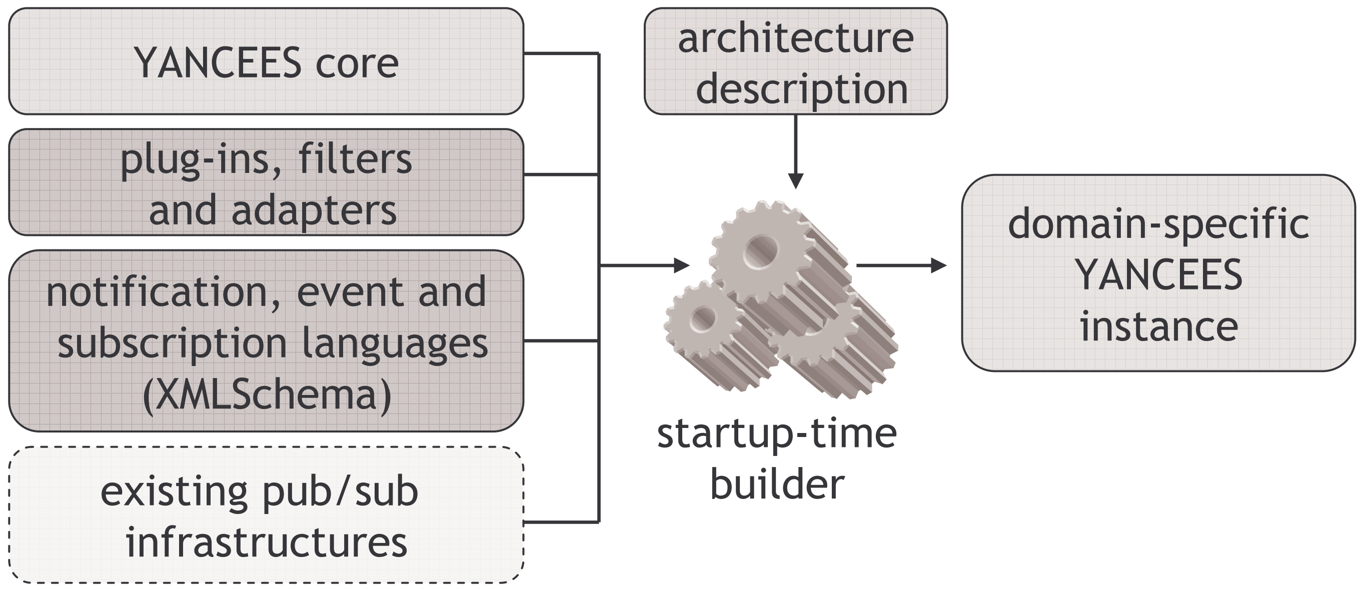
Based on the use of extensible languages, plug-ins and filters
- combining language and infrastructure evolution
- with static and dynamic plug-in configurations

- Built upon a micro kernel architecture style
  - achieving interoperability and support for different event models and routing strategies

- The architecture variability follows an extended version of Rosenblum and Wolf’s [24] publish/subscribe design dimensions

- The components are put together with the help of runtime parsers and static configuration managers

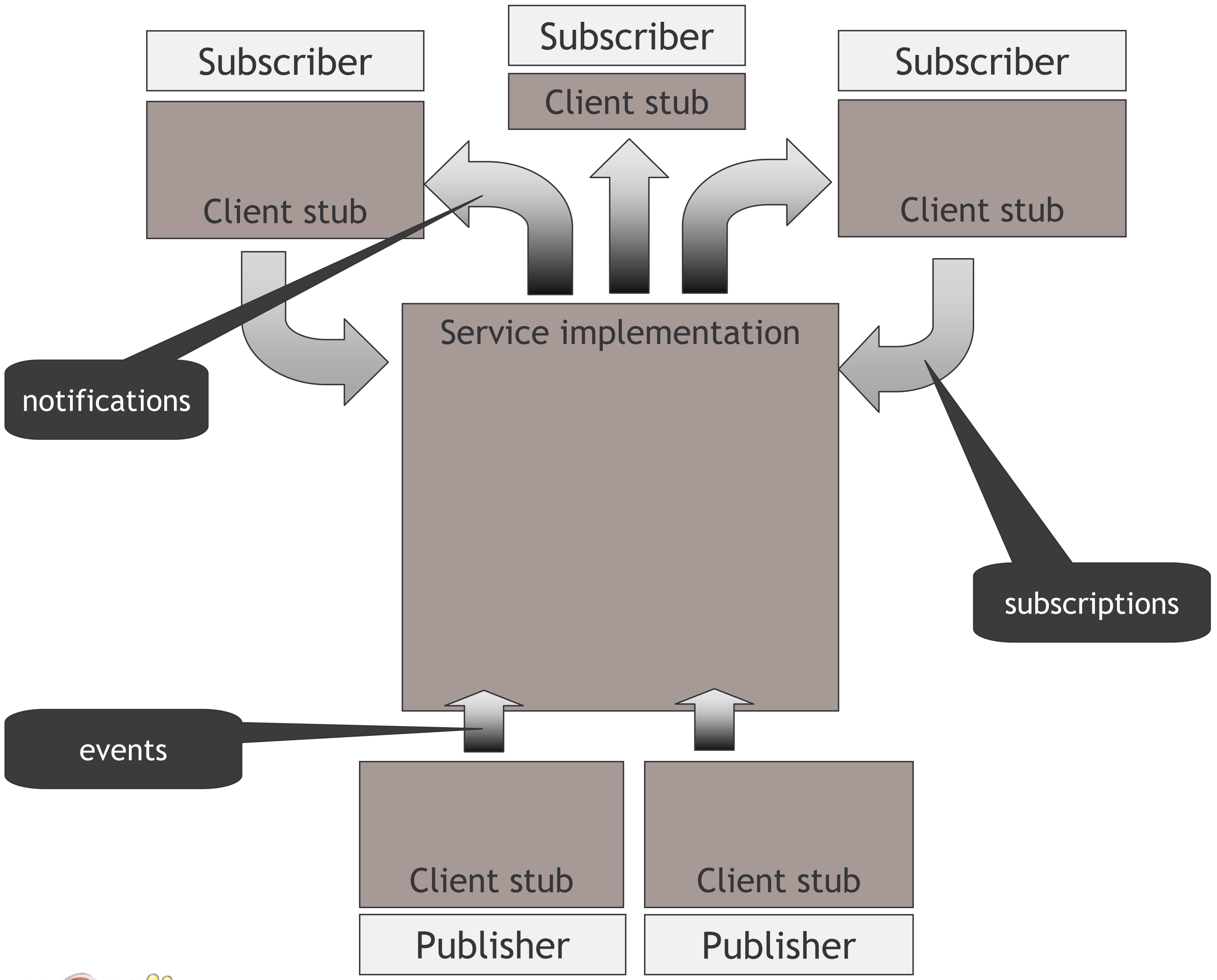
Approach summary

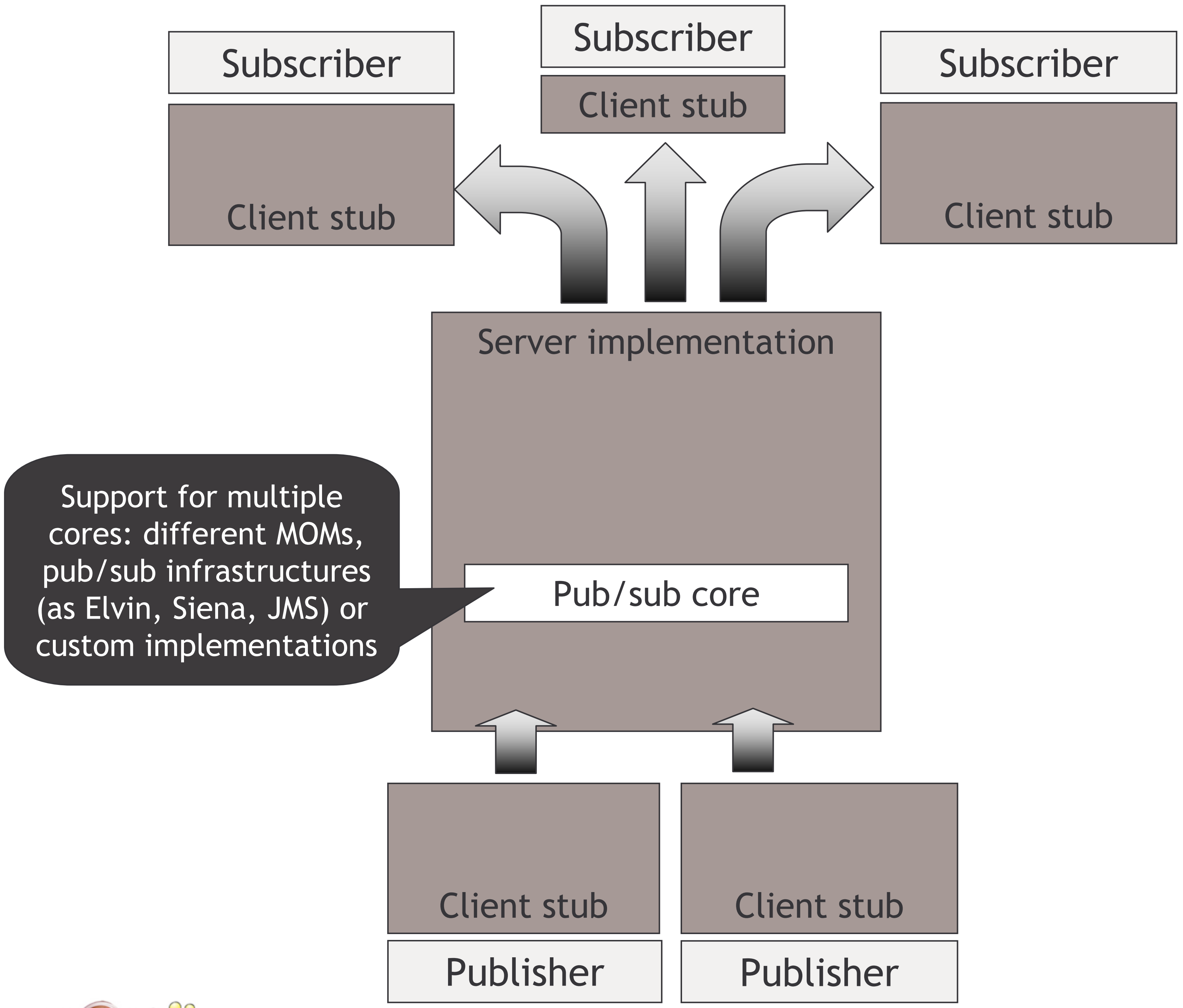

YANCEES core
plug-ins, filters and adapters
notification, event and subscription languages (XMLSchema)
existing pub/sub infrastructures
architecture description
startup-time builder
domain-specific YANCEES instance

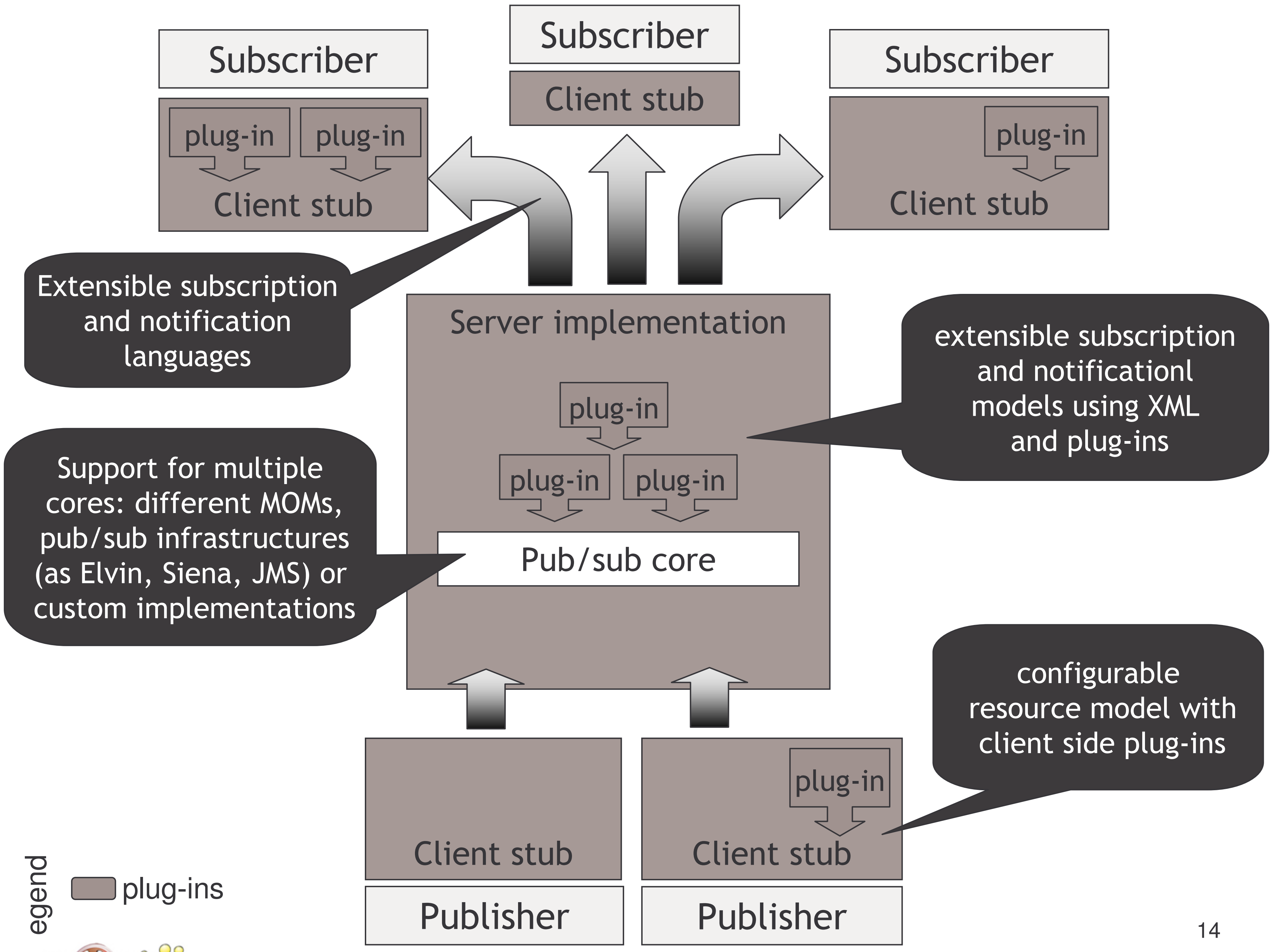
Publish/subscribe design dimensions

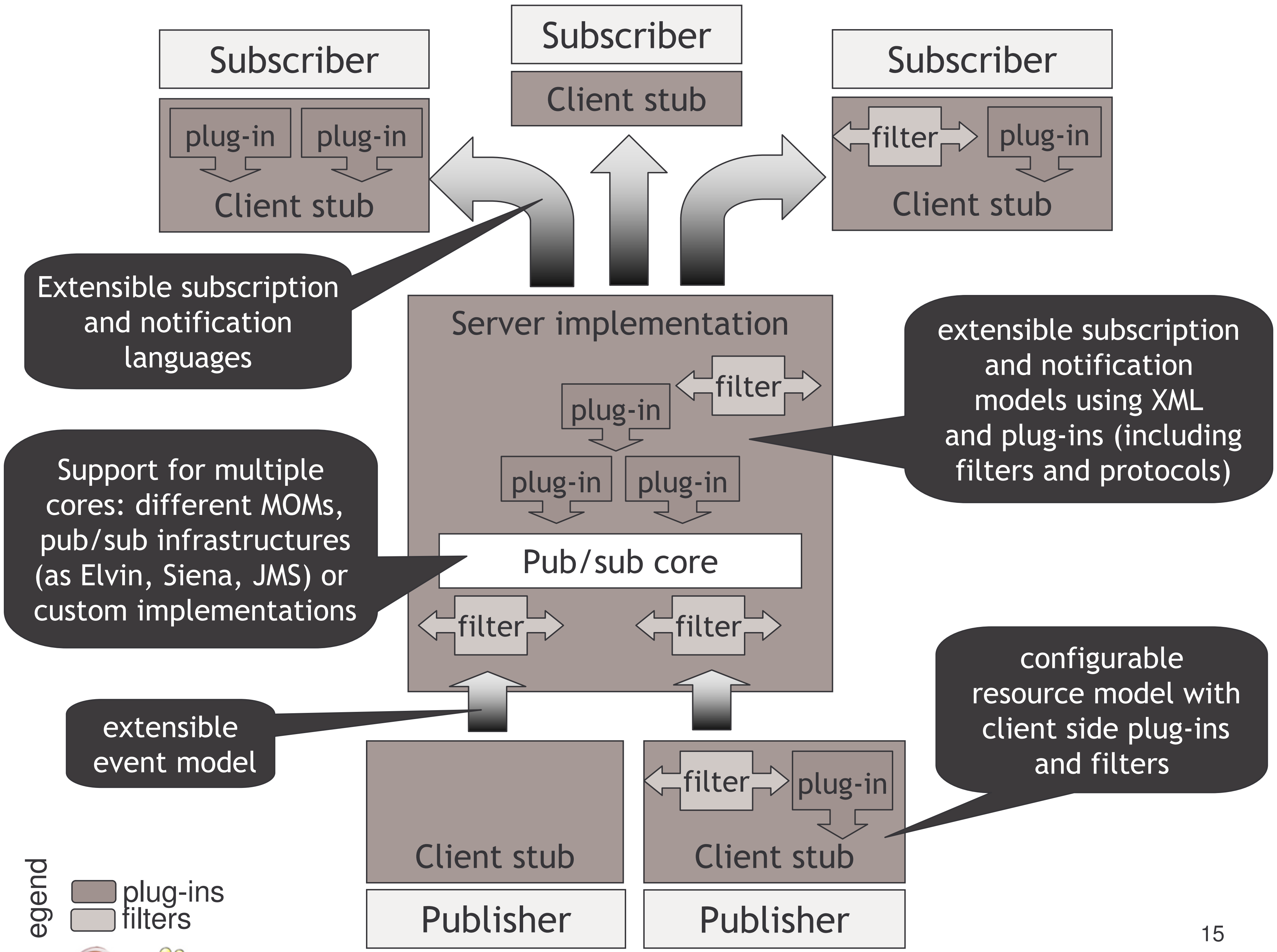
- **Extended version (see Protocol*) of Rosenblum and Wolf's model that represent the variability dimensions in our approach**

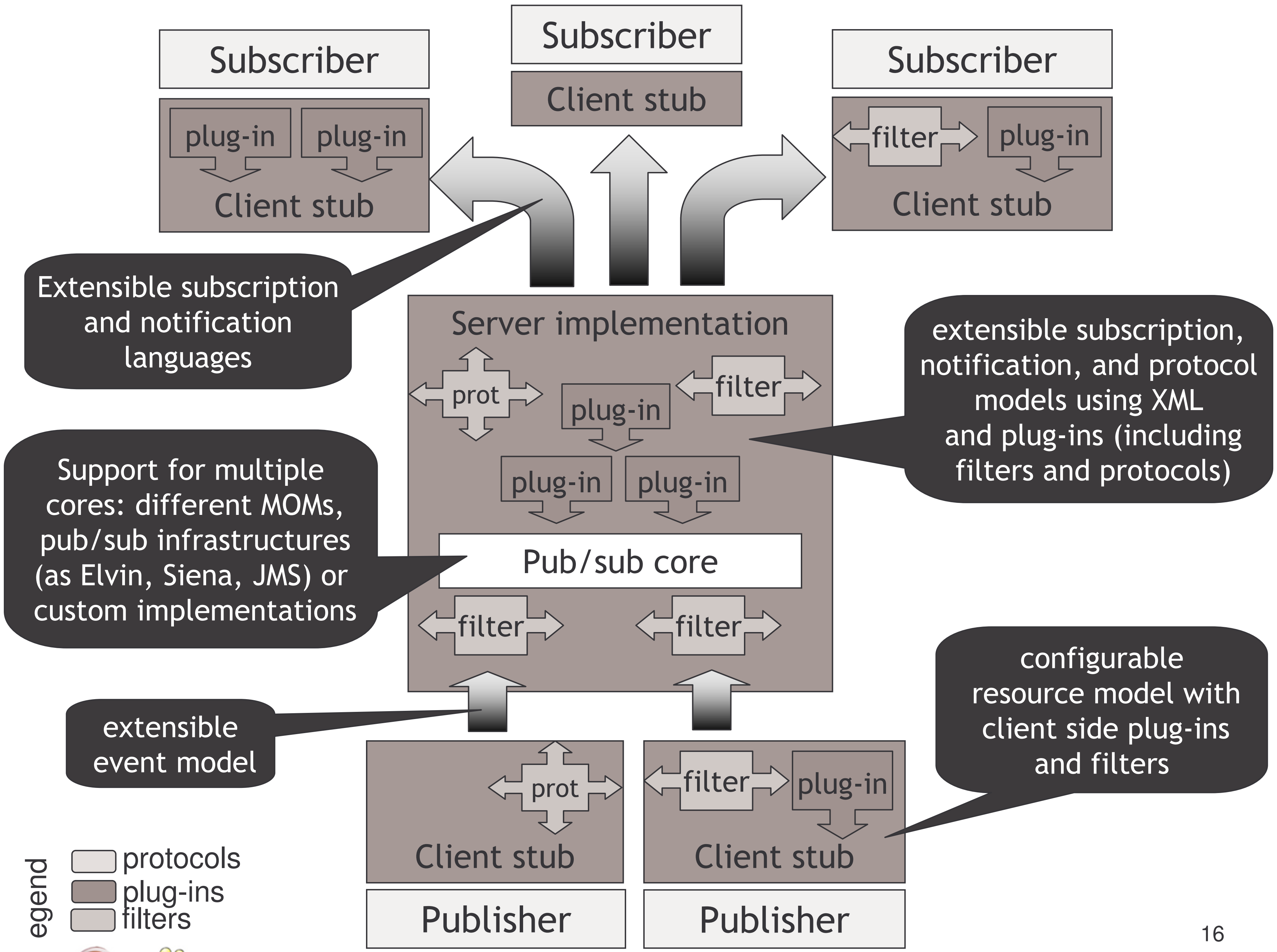
Dimension	Definition	Example
Subscription	specifies how subscribers express interest in subsets of events	content-based, topic-based, advanced event processing
Notification	specifies how notifications are delivered to subscribers	push, pull, both, others.
Event	Specifies how events are represented	tuple-based, record-based, XML documents
Protocol*	other kinds of interaction with the service	**Interaction protocols**: authentication, manual roaming **Infrastructure protocols**: federation, replication, fault-tolerance
Resource	defines where in the system (publishers/subscribers/routers) the extensions are placed	client-side, server-side


Subscriber
Client stub
Subscriber
Client stub
Subscriber
Client stub
notifications
Service implementation
subscriptions
events
Client stub
Publisher
Client stub
Publisher


Subscriber
Client stub
Subscriber
Client stub
Subscriber
Client stub
Server implementation
Pub/sub core
Support for multiple cores: different MOMs, pub/sub infrastructures (as Elvin, Siena, JMS) or custom implementations
Client stub
Publisher
Client stub
Publisher


Subscriber
plug-in
plug-in
Client stub
Subscriber
Client stub
Subscriber
plug-in
Client stub
Extensible subscription and notification languages
Server implementation
plug-in
plug-in
plug-in
Pub/sub core
extensible subscription and notificationl models using XML and plug-ins
Support for multiple cores: different MOMs, pub/sub infrastructures (as Elvin, Siena, JMS) or custom implementations
configurable resource model with client side plug-ins
Client stub
Publisher
plug-in
Client stub
Publisher
egend
plug-ins


Subscriber
plug-in
plug-in
Client stub
Subscriber
Client stub
Subscriber
filter
plug-in
Client stub
Extensible subscription and notification languages
Server implementation
filter
plug-in
plug-in
plug-in
Pub/sub core
filter
filter
extensible subscription and notification models using XML and plug-ins (including filters and protocols)
Support for multiple cores: different MOMs, pub/sub infrastructures (as Elvin, Siena, JMS) or custom implementations
extensible event model
configurable resource model with client side plug-ins and filters
Client stub
Publisher
filter
plug-in
Client stub
Publisher
egend
plug-ins
filters


Subscriber
Client stub
plug-in
plug-in
Subscriber
Client stub
Subscriber
filter
plug-in
Client stub
Extensible subscription and notification languages
Server implementation
prot
plug-in
filter
plug-in
plug-in
Pub/sub core
filter
filter
extensible subscription, notification, and protocol models using XML and plug-ins (including filters and protocols)
Support for multiple cores: different MOMs, pub/sub infrastructures (as Elvin, Siena, JMS) or custom implementations
configurable resource model with client side plug-ins and filters
extensible event model
prot
Client stub
filter
plug-in
Client stub
Publisher
Publisher
egend
protocols
plug-ins
filters

Variability dimensions summary

Dimension	Approach
Subscription	Extensible subscription language Subscription plug-ins
Notification	Extensible notification language Notification plug-ins
Event	Extensible event representation Filters Event adapters and publish/subscribe cores
Protocol	Protocol plug-ins
Resource	Configuration managers that interpret configuration descriptions Dynamic parsers

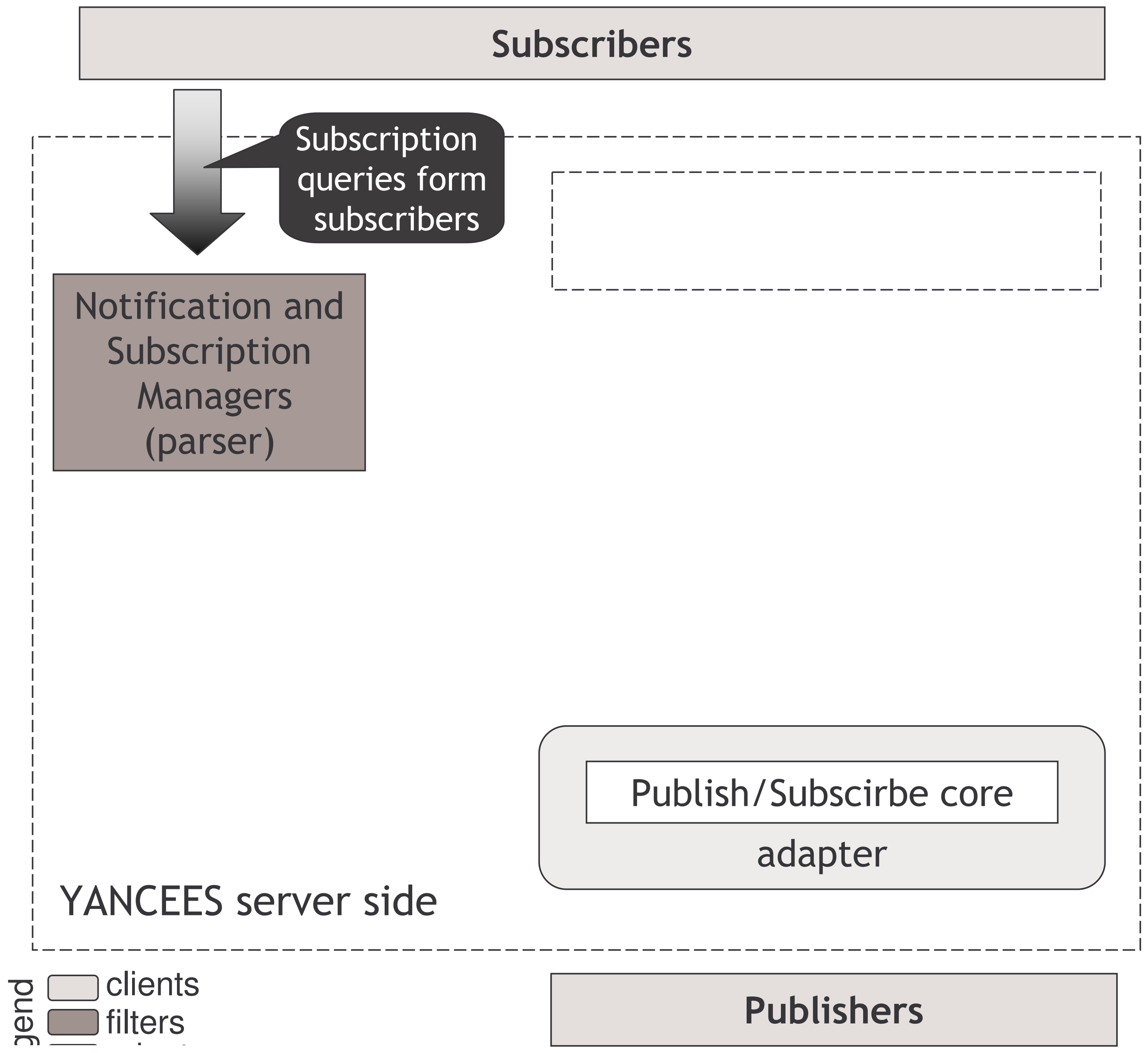
Implementation details

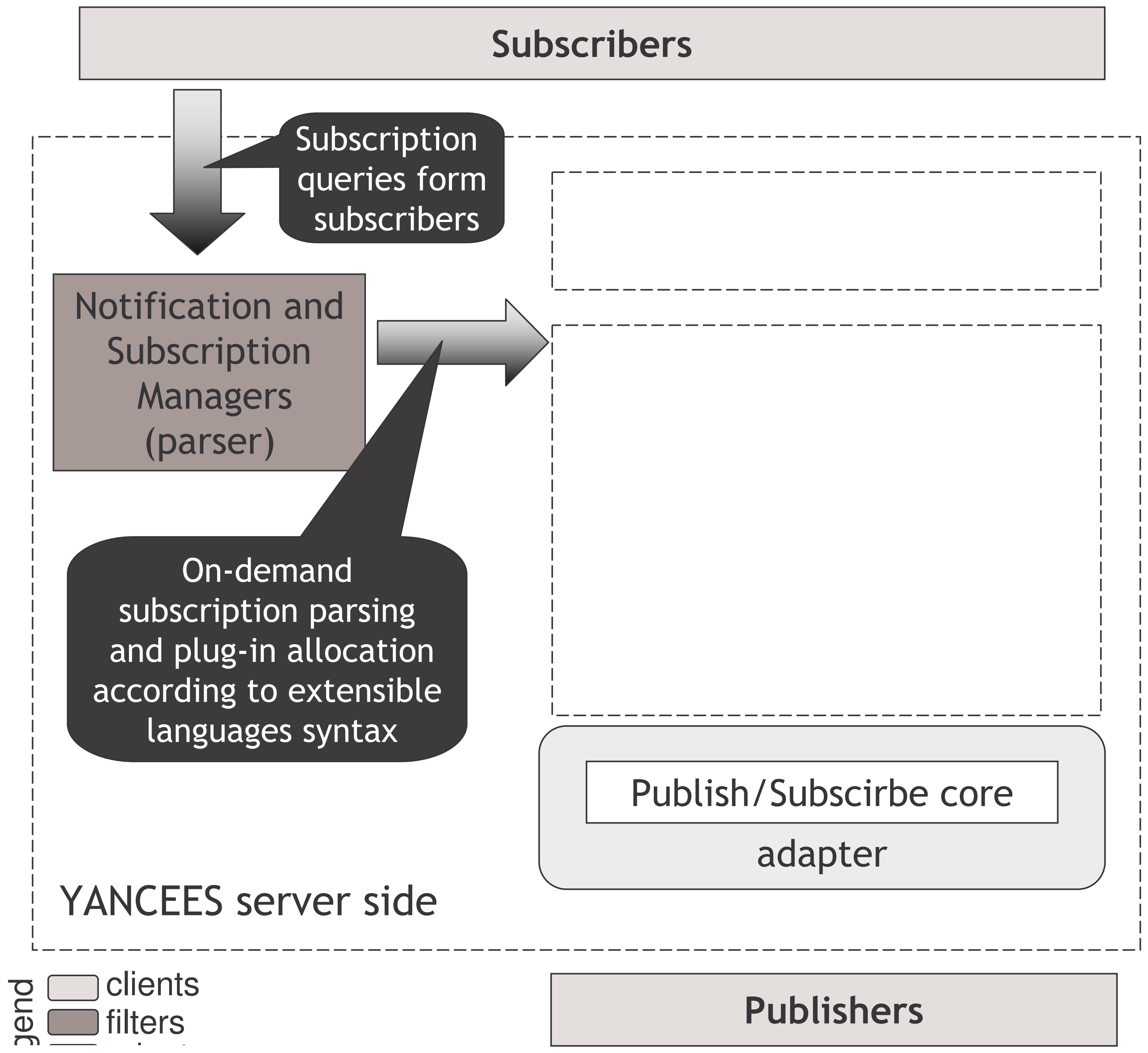
- The application is in Java
- The extensible language used is XML (XMLSchema)
- Events, Subscriptions and Notifications are all represented in XML, as well as the configuration language.
  - Events can also be objects.
- The interaction with the service (pub/sub API) is done through RMI
- Protocol plug-in interfaces are currently using RMI
- Siena, Elvin and a custom topic-based switcher were used as the basic pub/sub cores

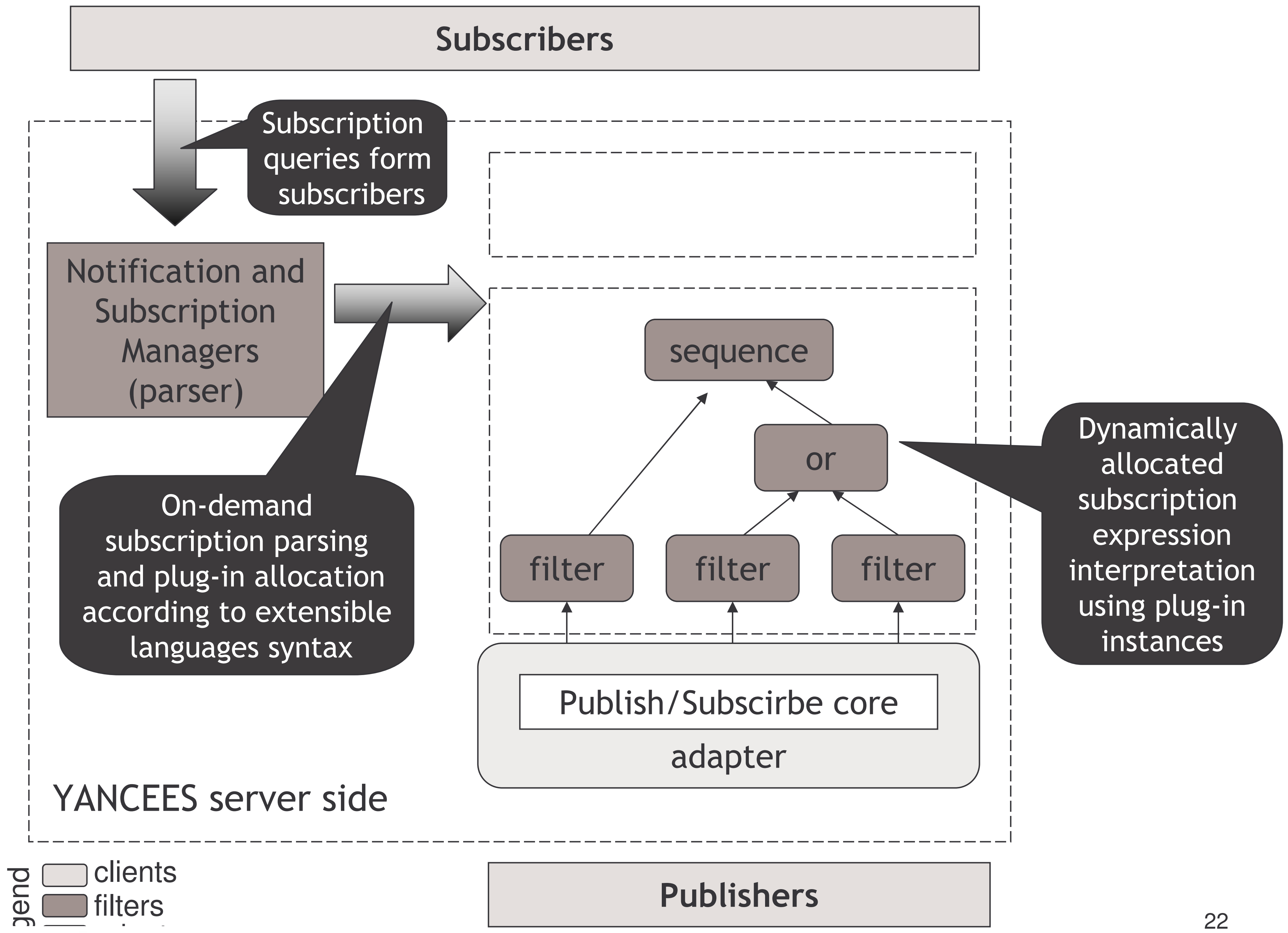
Example of dynamic parsing of subscriptions

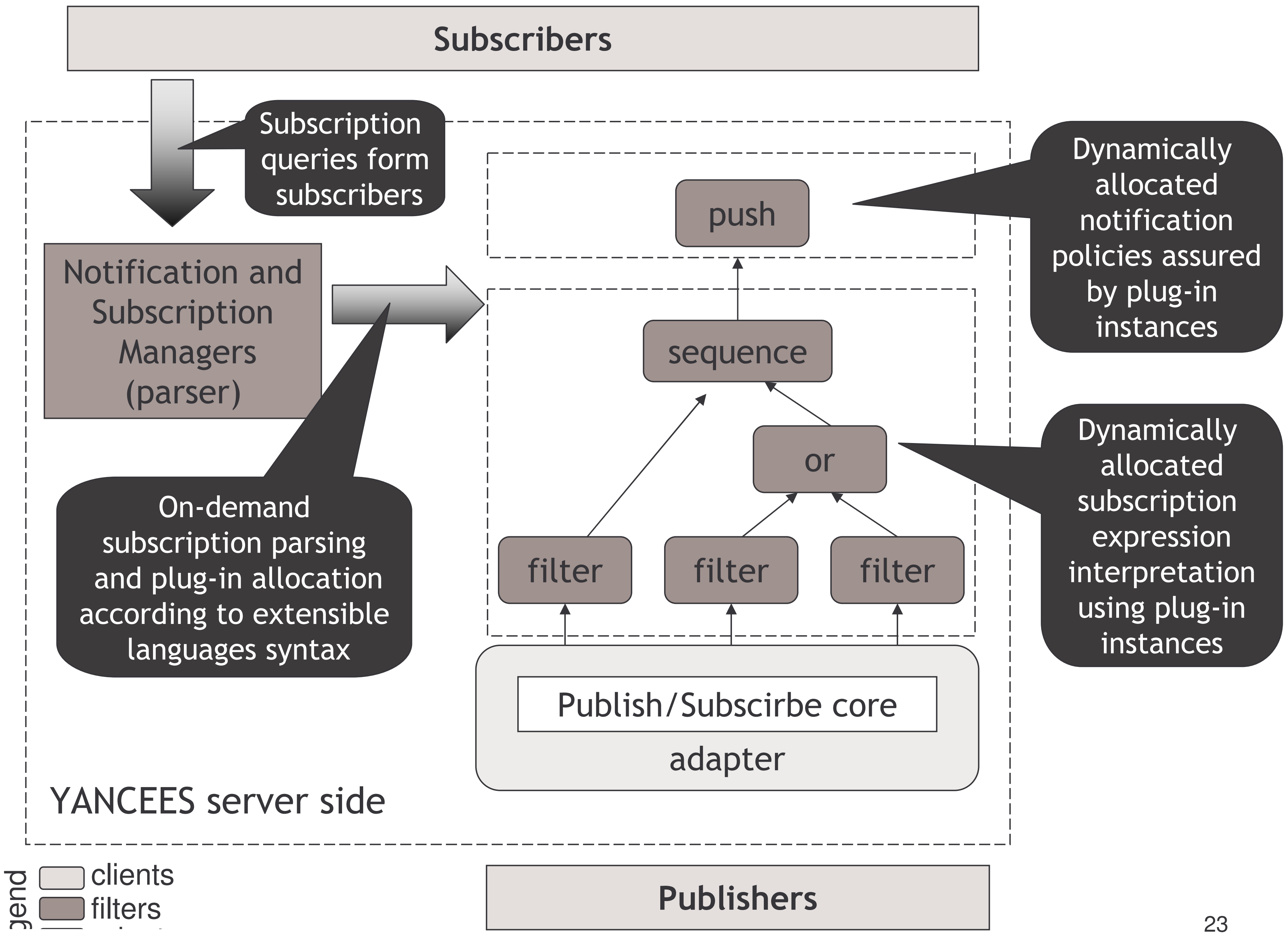
- Consider the following sequence detection subscription as an example:

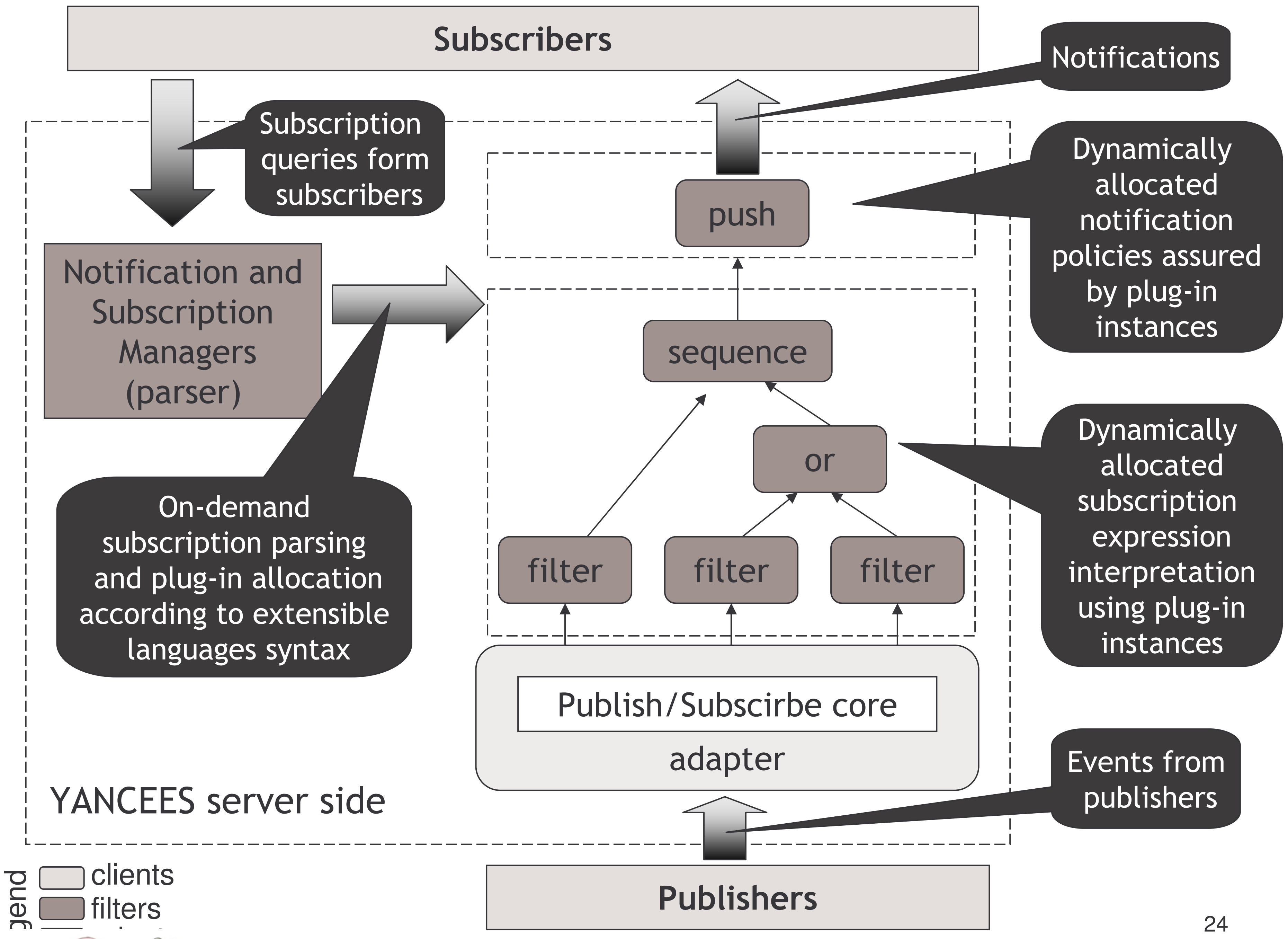
```
<subscribe>
  <sequence>
    <or>
      <filter>
       <eq> <name> name </name>
            <value> Roberto </name> </eq>
      </filter>
        <filter> … </filter>
    </or>
    <filter> … </filter>
  </sequence>
  <push>
</subscribe>
```


Subscribers
Subscription queries form subscribers
Notification and Subscription Managers (parser)
Publish/Subscirbe core
adapter
YANCEES server side
Publishers
gend
clients
filters

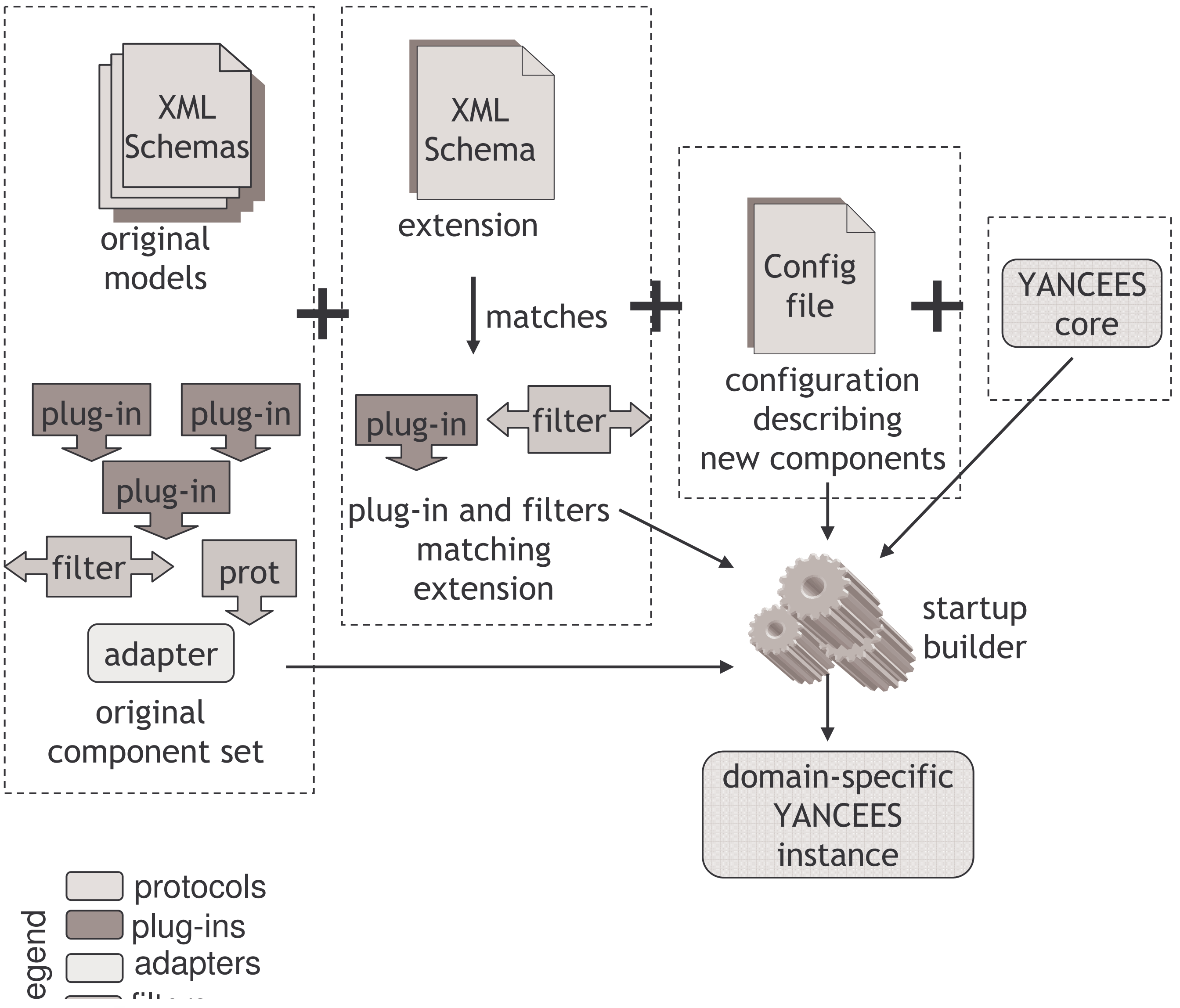

Subscribers
Subscription queries form subscribers
Notification and Subscription Managers (parser)
On-demand subscription parsing and plug-in allocation according to extensible languages syntax
Publish/Subscirbe core
adapter
YANCEES server side
Publishers
gend
clients
filters


Subscribers
Subscription queries form subscribers
Notification and Subscription Managers (parser)
On-demand subscription parsing and plug-in allocation according to extensible languages syntax
sequence
or
filter
filter
filter
Dynamically allocated subscription expression interpretation using plug-in instances
Publish/Subscirbe core
adapter
YANCEES server side
Publishers
clients
filters


Subscribers
Subscription queries form subscribers
Notification and Subscription Managers (parser)
On-demand subscription parsing and plug-in allocation according to extensible languages syntax
push
sequence
or
filter
filter
filter
Dynamically allocated notification policies assured by plug-in instances
Dynamically allocated subscription expression interpretation using plug-in instances
Publish/Subscirbe core
adapter
YANCEES server side
Publishers
clients
filters


Subscribers
Notifications
Subscription queries form subscribers
Dynamically allocated notification policies assured by plug-in instances
push
Notification and Subscription Managers (parser)
sequence
or
Dynamically allocated subscription expression interpretation using plug-in instances
On-demand subscription parsing and plug-in allocation according to extensible languages syntax
filter
filter
filter
Publish/Subscirbe core
adapter
Events from publishers
YANCEES server side
Publishers
gend
clients
filters

How to extend YANCEES?



1. Define a language extension using XML Schema
2. Implement plug-ins and filters for that extension
3. register plug-in in the configuration file
4. Restart the server with the new configuration

How does it compare to other approaches? (e.g. Elvin, Siena or CORBA-NS)

- None of them are programmable. And they are not easily extensible.
- Siena and Elvin, for e.g., provide content-based routing and sequence detection with push notification only
- CORBA-NS allows the selection of notification (push, pull) and subscription policies to use (channel, topic). The event model is fixed (object-based) at runtime.
  - It is monolithic and no additional features can not be easily added to it or removed from it.

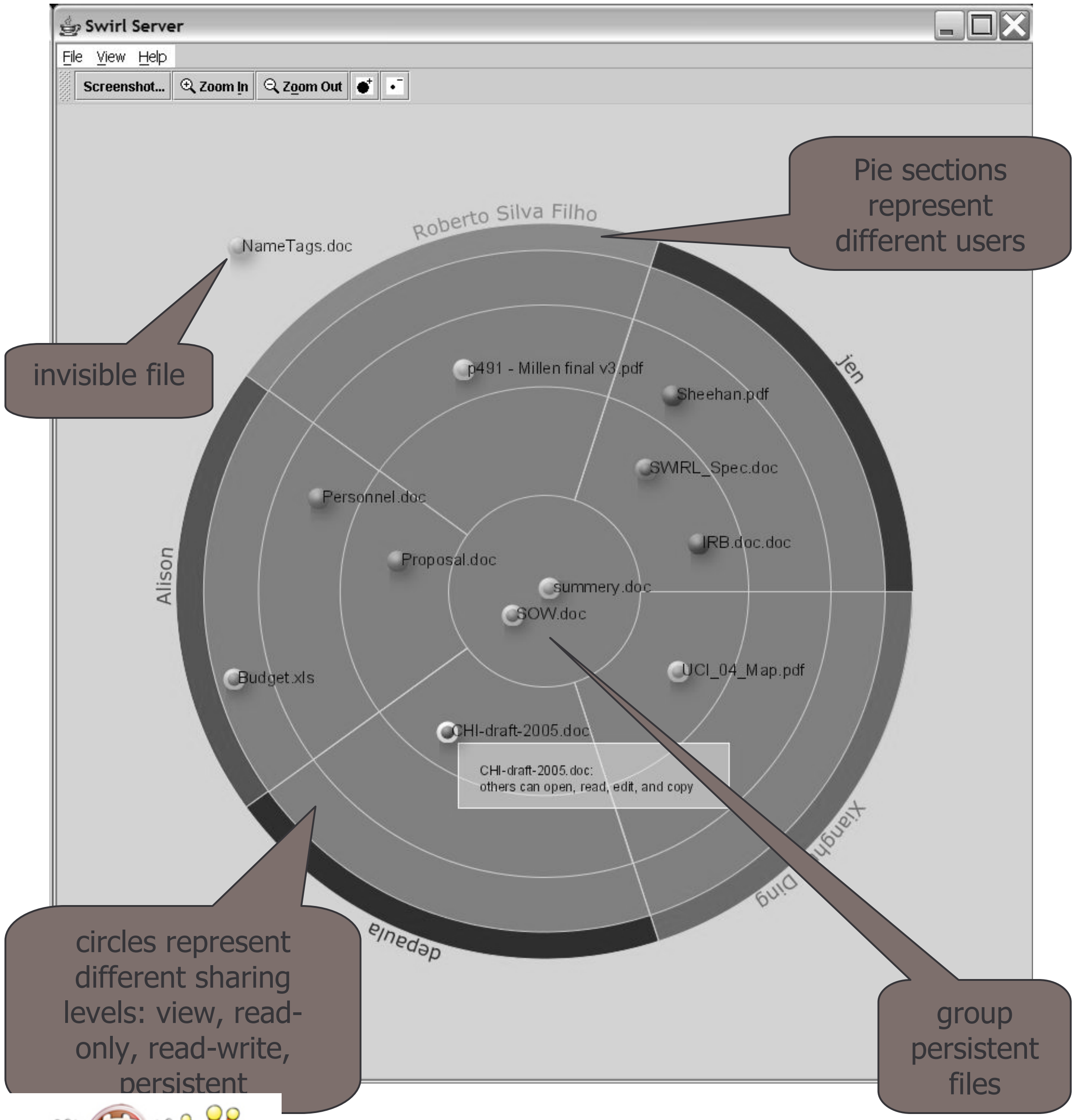
Three example applications and their configurations

- Implementation of a peer-to-peer event bus for ad-hoc file sharing application
- support for a software visualization tool and network activity monitoring application
- Implementation of an awareness server (CASSIUS equivalent)

Peer-to-peer file sharing tool

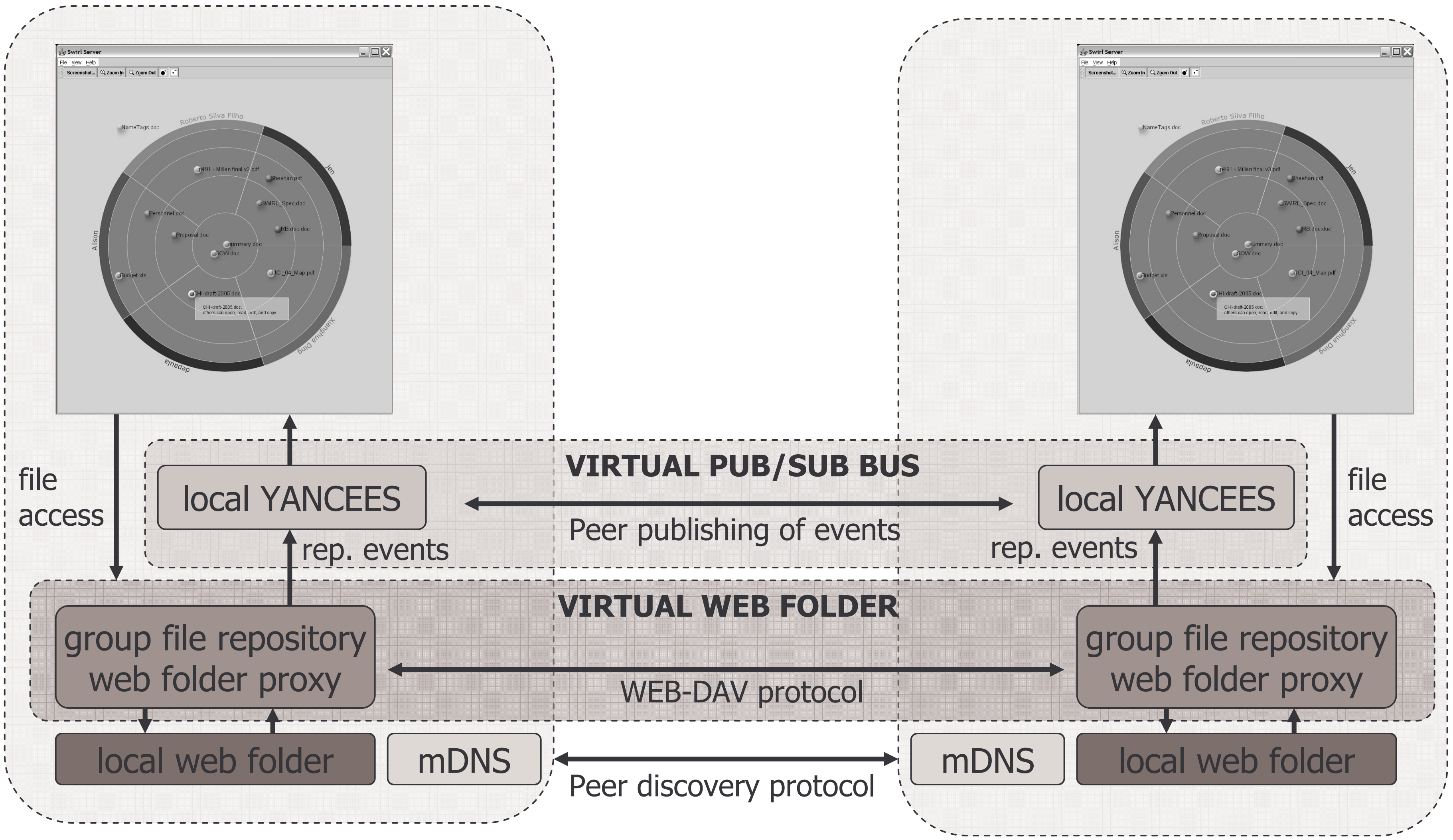
- YANCEES is used to provide a P2P event bus that supports:
  - dynamic peer discovery
  - peer-to-peer publishing

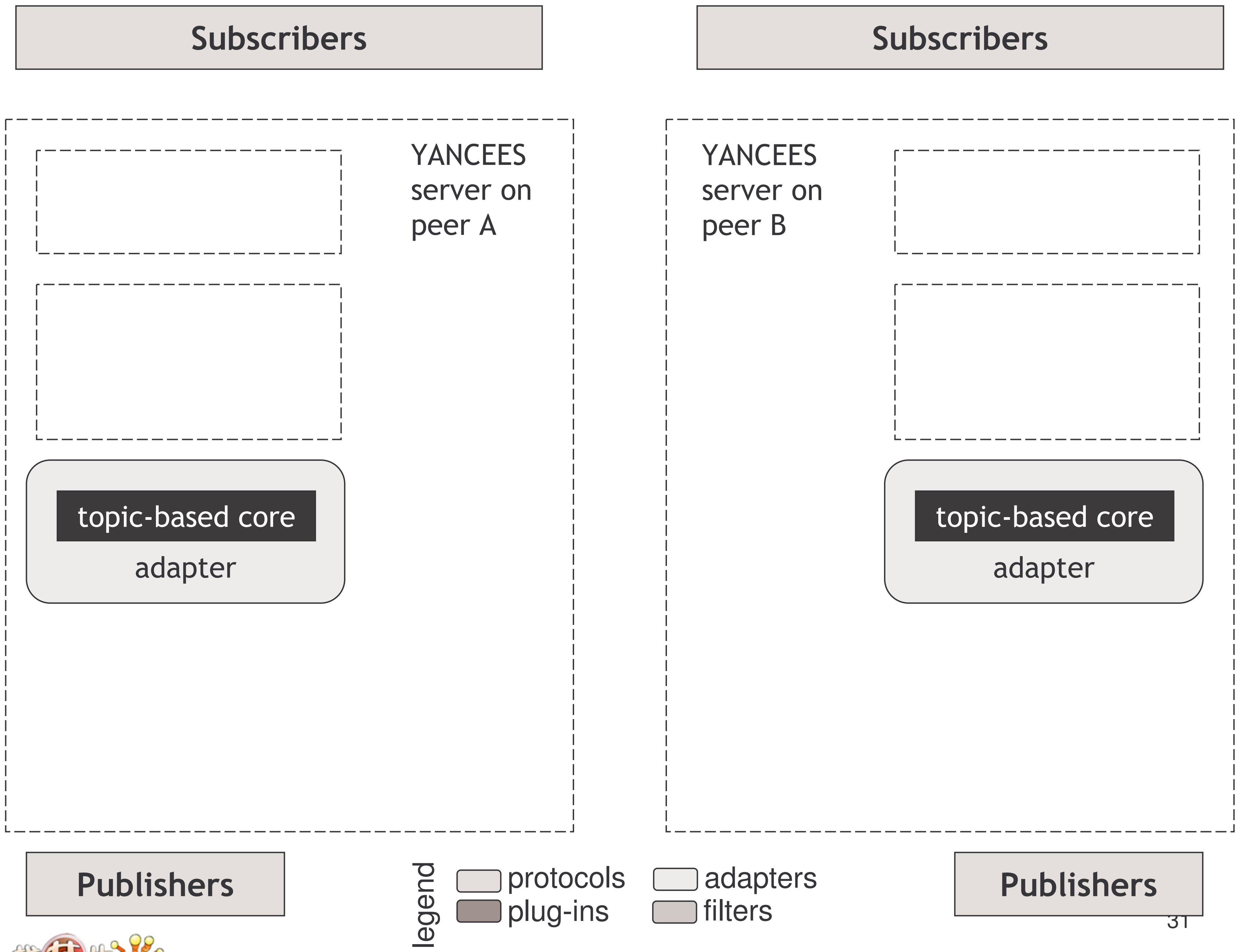
Impromptu: a peer-to-peer, ad-hoc, file sharing application

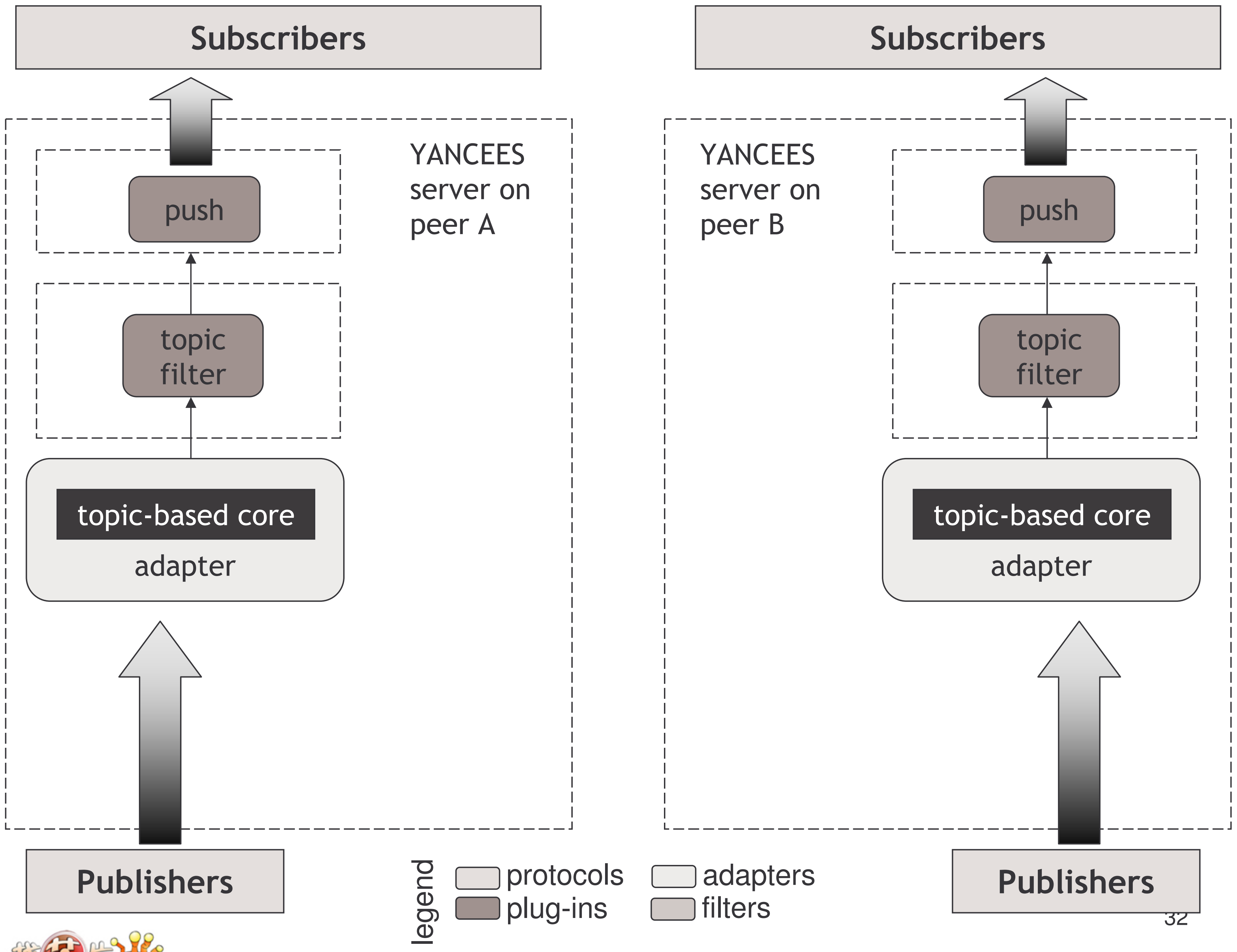


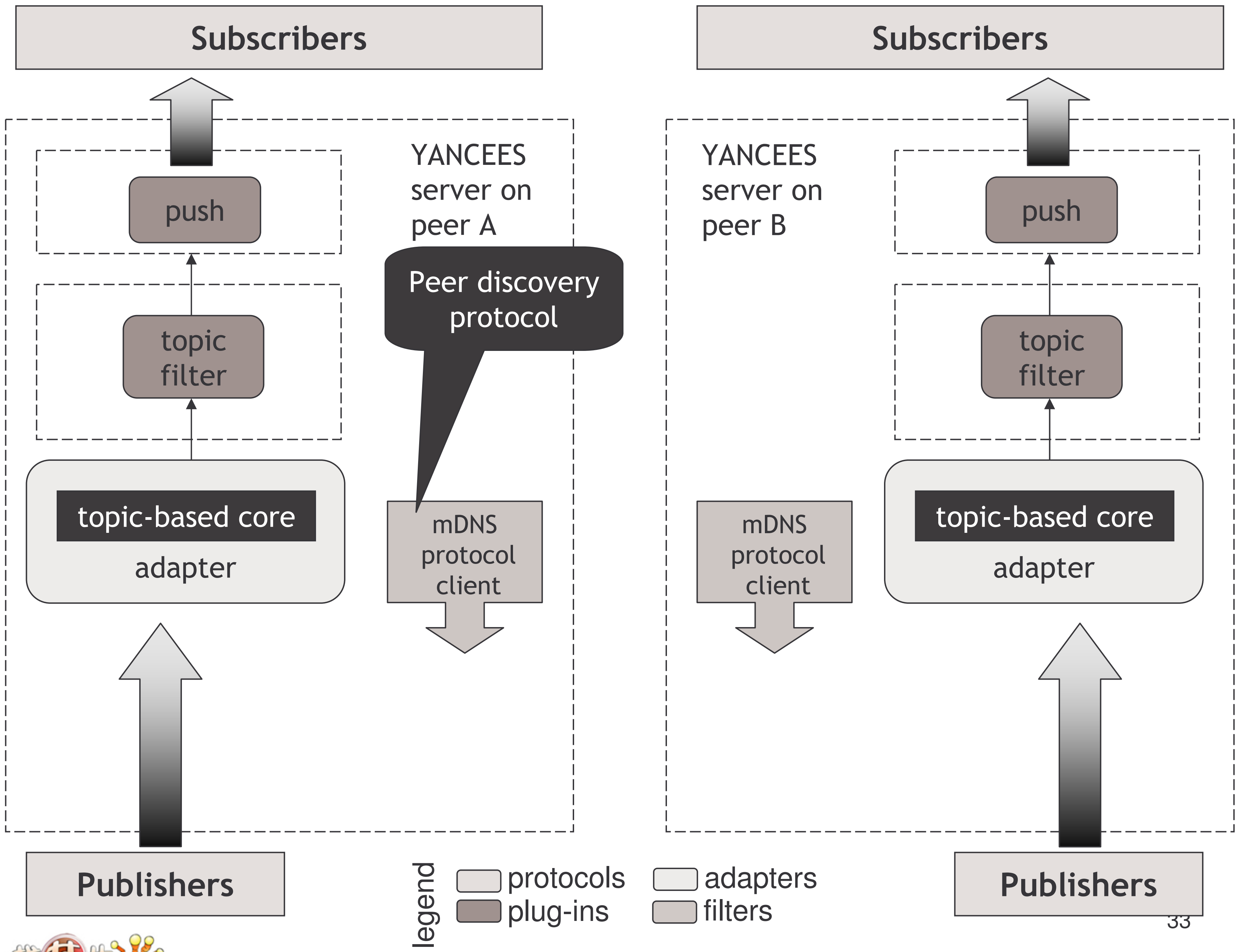
- Different pie slices represent peers in the network
- Users can drag and drop files to be shared
- Concentric circles define different sharing levels
- Files placed in the center are persistent and available to all the members of the group
- Files outside the pie are not shared and become invisible to other peers
- Files blink with the peer color whenever someone reads or modifies it

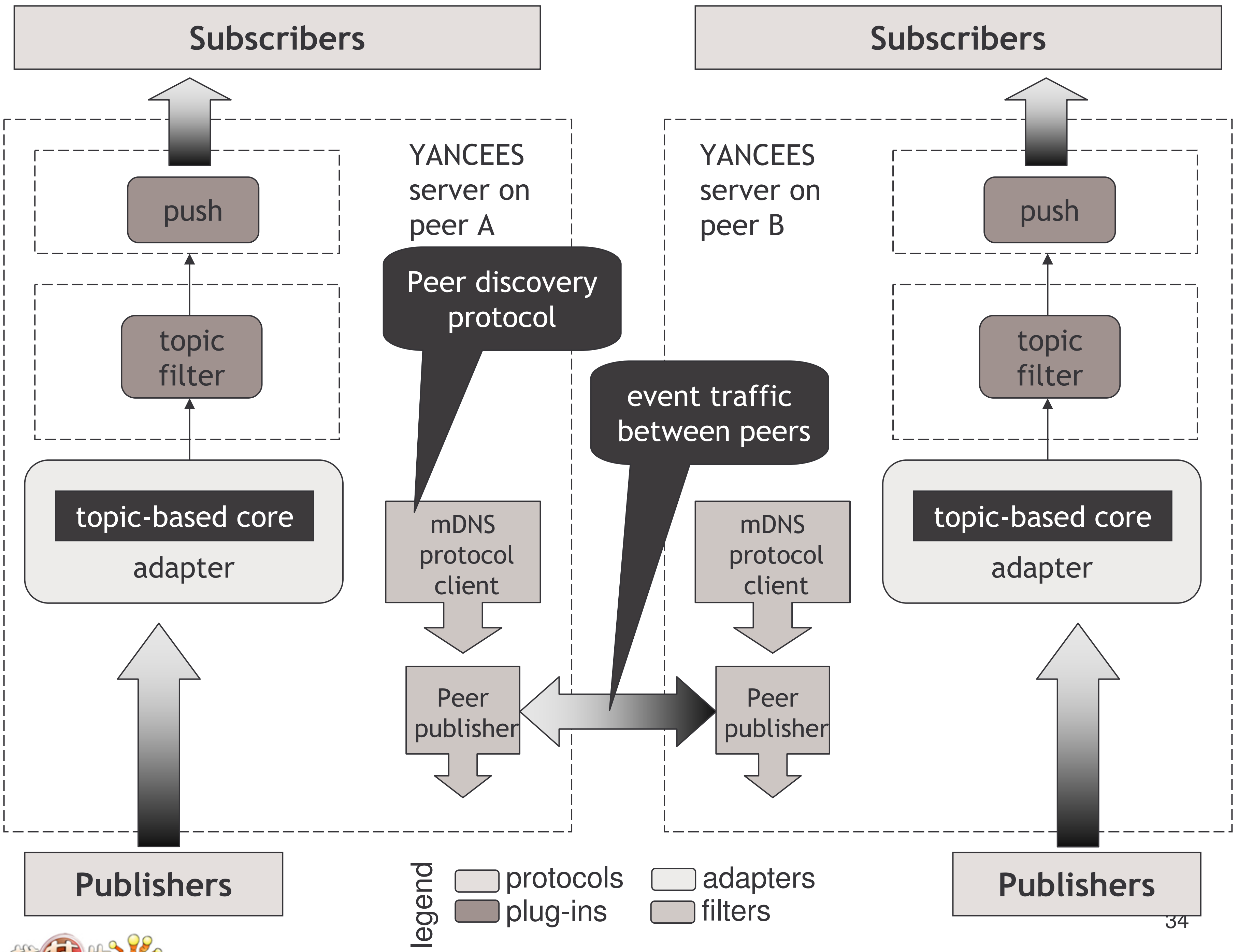
Impromptu architecture: peer-to-peer file sharing tool support

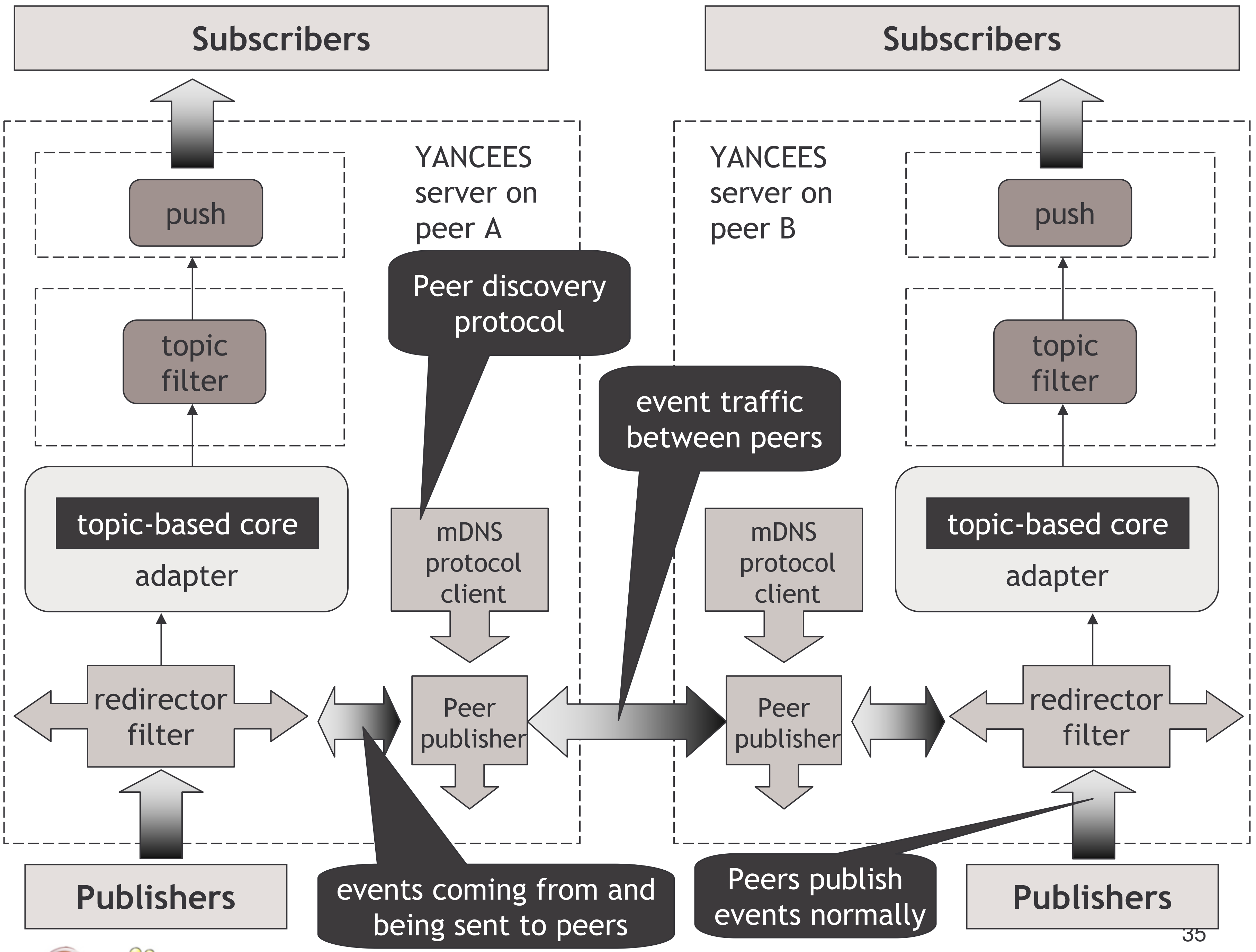



Subscribers
Subscribers
YANCEES server on peer A
YANCEES server on peer B
topic-based core
adapter
topic-based core
adapter
Publishers
Publishers
legend
protocols
plug-ins
adapters
filters


Subscribers
YANCEES server on peer A
push
topic filter
topic-based core
adapter
Publishers
Subscribers
YANCEES server on peer B
push
topic filter
topic-based core
adapter
Publishers
legend
protocols
plug-ins
adapters
filters


Subscribers
Subscribers
YANCEES server on peer A
YANCEES server on peer B
push
push
topic filter
topic filter
Peer discovery protocol
topic-based core
adapter
topic-based core
adapter
mDNS protocol client
mDNS protocol client
Publishers
Publishers
legend
protocols
plug-ins
adapters
filters

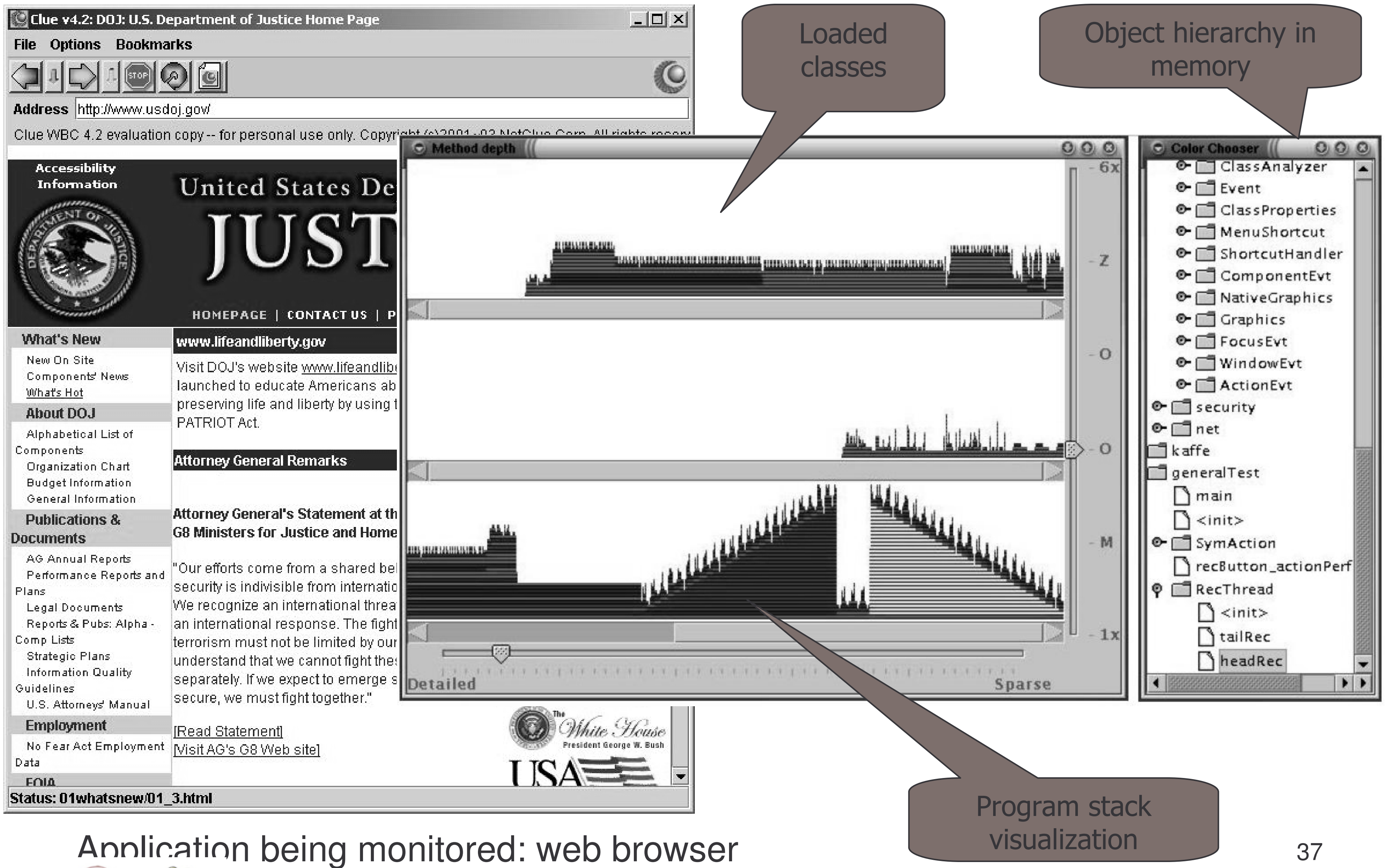

Subscribers
Subscribers
YANCEES server on peer A
YANCEES server on peer B
push
push
topic filter
topic filter
Peer discovery protocol
event traffic between peers
topic-based core
adapter
topic-based core
adapter
mDNS protocol client
mDNS protocol client
Peer publisher
Peer publisher
Publishers
Publishers
legend
protocols
plug-ins
adapters
filters


Subscribers
Subscribers
YANCEES server on peer A
YANCEES server on peer B
push
push
topic filter
topic filter
Peer discovery protocol
event traffic between peers
topic-based core
adapter
topic-based core
adapter
mDNS protocol client
mDNS protocol client
redirector filter
Peer publisher
Peer publisher
redirector filter
Publishers
events coming from and being sent to peers
Peers publish events normally
Publishers

Software and security visualization

- YANCEES addresses two different routing requirements:
  - – Software visualization: support fast routing
  - – Security visualization: content-based filtering

Software visualization tool



Clue v4.2: DOJ: U.S. Department of Justice Home Page

http://www.usdoj.gov/

Clue WBC 4.2 evaluation copy -- for personal use only. Copyright (c)2001-03 NetClue Corp. All rights reserv

Accessibility Information

United States De

JUST

HOMEPAGE | CONTACT US | P

What's New
New On Site
Components' News
What's Hot

About DOJ
Alphabetical List of Components
Organization Chart
Budget Information
General Information

Publications & Documents
AG Annual Reports
Performance Reports and Plans
Legal Documents
Reports & Pubs: Alpha - Comp Lists
Strategic Plans
Information Quality Guidelines
U.S. Attorneys' Manual

Employment
No Fear Act Employment Data

FOIA

www.lifeandliberty.gov

Visit DOJ's website www.lifeandlib
launched to educate Americans ab
preserving life and liberty by using t
PATRIOT Act.

Attorney General Remarks

Attorney General's Statement at th
G8 Ministers for Justice and Home

"Our efforts come from a shared bel
security is indivisible from internatio
We recognize an international threa
an international response. The fight
terrorism must not be limited by our
understand that we cannot fight the
separately. If we expect to emerge s
secure, we must fight together."

[Read Statement]
[Visit AG's G8 Web site]

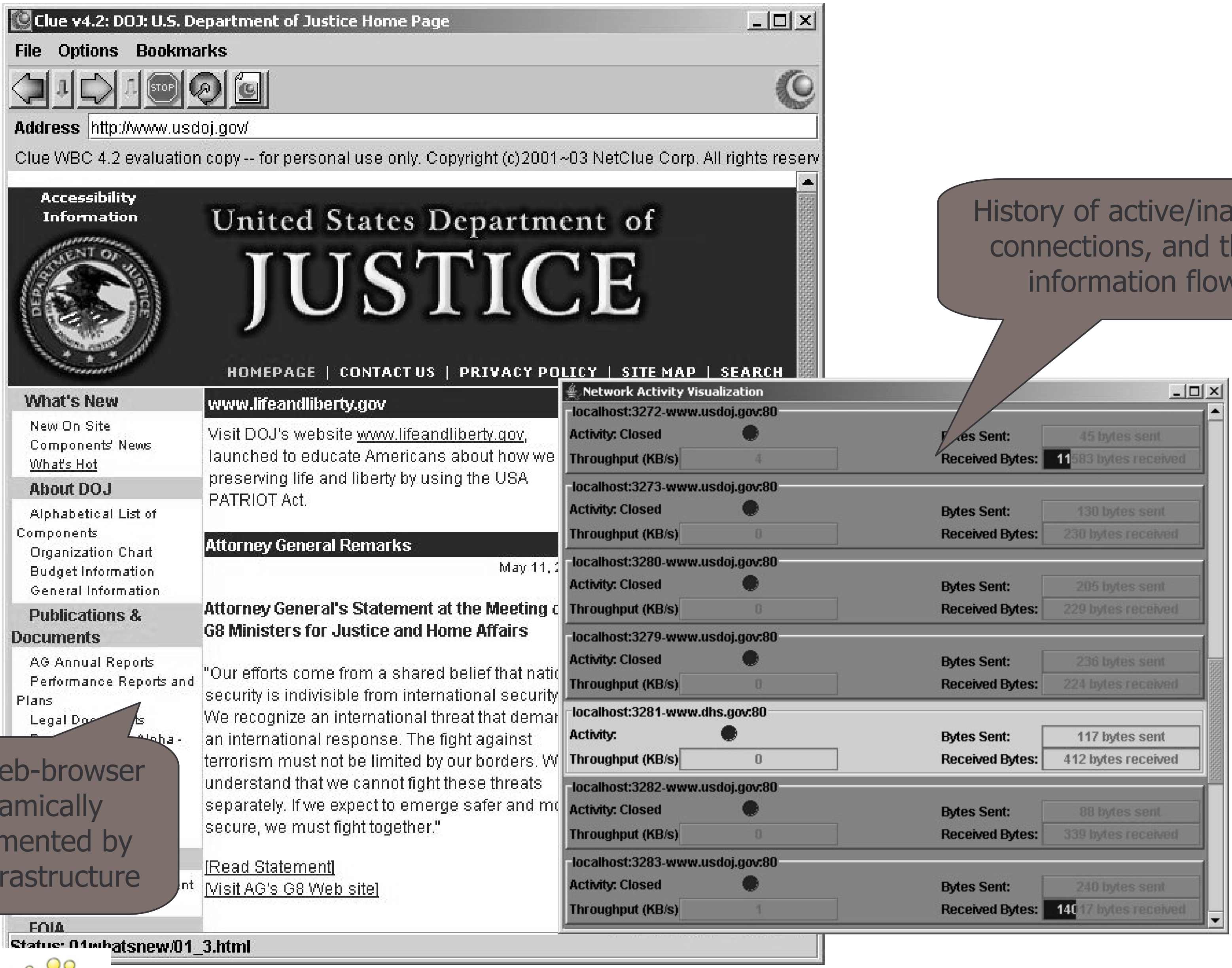
The White House
President George W. Bush

USA

Status: 01whatsnew/01_3.html

Application being monitored: web browser

Security visualization tool



Clue v4.2: DOJ: U.S. Department of Justice Home Page

File Options Bookmarks

Address http://www.usdoj.gov/

Clue WBC 4.2 evaluation copy -- for personal use only. Copyright (c)2001~03 NetClue Corp. All rights reserv

Accessibility Information

United States Department of JUSTICE

HOMEPAGE | CONTACT US | PRIVACY POLICY | SITE MAP | SEARCH

What's New
New On Site
Components' News
What's Hot

About DOJ
Alphabetical List of Components
Organization Chart
Budget Information
General Information

Publications & Documents
AG Annual Reports
Performance Reports and Plans
Legal Do

FOIA

www.lifeandliberty.gov

Visit DOJ's website www.lifeandliberty.gov, launched to educate Americans about how we preserving life and liberty by using the USA PATRIOT Act.

Attorney General Remarks

May 11,

Attorney General's Statement at the Meeting c G8 Ministers for Justice and Home Affairs

"Our efforts come from a shared belief that natic security is indivisible from international security We recognize an international threat that demar an international response. The fight against terrorism must not be limited by our borders. W understand that we cannot fight these threats separately. If we expect to emerge safer and mc secure, we must fight together."

[Read Statement]
[Visit AG's G8 Web site]

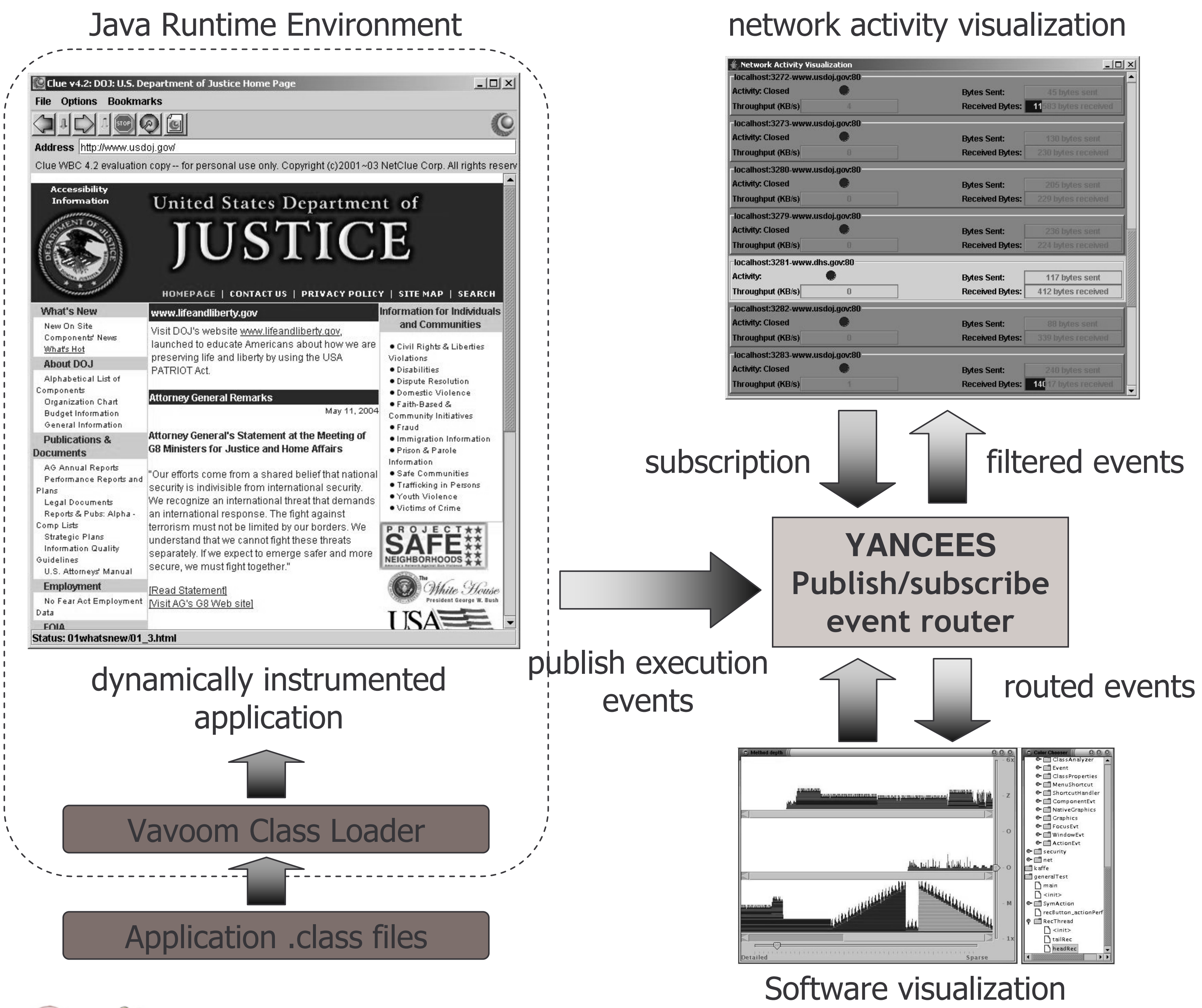
Status: 01whatsnew/01_3.html

Network Activity Visualization

```
localhost:3272-www.usdoj.gov:80
Activity: Closed                      Bytes Sent:      45 bytes sent
Throughput (KB/s)  4                  Received Bytes:  11583 bytes received
localhost:3273-www.usdoj.gov:80
Activity: Closed                      Bytes Sent:      130 bytes sent
Throughput (KB/s)  0                  Received Bytes:  230 bytes received
localhost:3280-www.usdoj.gov:80
Activity: Closed                      Bytes Sent:      205 bytes sent
Throughput (KB/s)  0                  Received Bytes:  229 bytes received
localhost:3279-www.usdoj.gov:80
Activity: Closed                      Bytes Sent:      236 bytes sent
Throughput (KB/s)  0                  Received Bytes:  224 bytes received
localhost:3281-www.dhs.gov:80
Activity:                             Bytes Sent:      117 bytes sent
Throughput (KB/s)  0                  Received Bytes:  412 bytes received
localhost:3282-www.usdoj.gov:80
Activity: Closed                      Bytes Sent:      88 bytes sent
Throughput (KB/s)  0                  Received Bytes:  339 bytes received
localhost:3283-www.usdoj.gov:80
Activity: Closed                      Bytes Sent:      240 bytes sent
Throughput (KB/s)  1                  Received Bytes:  14017 bytes received
```

History of active/inactive connections, and their information flow

Java web-browser dynamically instrumented by our infrastructure



Clue v4.2: DOJ: U.S. Department of Justice Home Page

File Options Bookmarks

Address http://www.usdoj.gov/

Clue WBC 4.2 evaluation copy -- for personal use only. Copyright (c)2001~03 NetClue Corp. All rights reserv

Accessibility Information

United States Department of JUSTICE

HOMEPAGE | CONTACT US | PRIVACY POLICY | SITE MAP | SEARCH

What's New
New On Site
Components' News
What's Hot

About DOJ
Alphabetical List of Components
Organization Chart
Budget Information
General Information

Publications & Documents
AG Annual Reports
Performance Reports and Plans
Legal Documents
Reports & Pubs: Alpha - Comp Lists
Strategic Plans
Information Quality Guidelines
U.S. Attorneys' Manual

Employment
No Fear Act Employment Data

FOIA

www.lifeandliberty.gov

Visit DOJ's website www.lifeandliberty.gov, launched to educate Americans about how we are preserving life and liberty by using the USA PATRIOT Act.

Attorney General Remarks

May 11, 2004

Attorney General's Statement at the Meeting of G8 Ministers for Justice and Home Affairs

"Our efforts come from a shared belief that national security is indivisible from international security. We recognize an international threat that demands an international response. The fight against terrorism must not be limited by our borders. We understand that we cannot fight these threats separately. If we expect to emerge safer and more secure, we must fight together."

[Read Statement]
[Visit AG's G8 Web site]

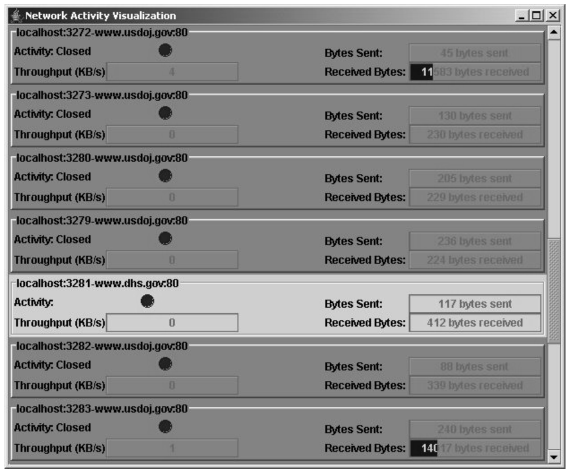
Information for Individuals and Communities
- Civil Rights & Liberties Violations
- Disabilities
- Dispute Resolution
- Domestic Violence
- Faith-Based & Community Initiatives
- Fraud
- Immigration Information
- Prison & Parole Information
- Safe Communities
- Trafficking in Persons
- Youth Violence
- Victims of Crime

PROJECT SAFE NEIGHBORHOODS

The White House President George W. Bush

USA

Status: 01whatsnew/01_3.html


Network Activity Visualization

localhost:3272-www.usdoj.gov:80
Activity: Closed — Bytes Sent: 45 bytes sent
Throughput (KB/s) 4 — Received Bytes: 11583 bytes received

localhost:3273-www.usdoj.gov:80
Activity: Closed — Bytes Sent: 130 bytes sent
Throughput (KB/s) 0 — Received Bytes: 230 bytes received

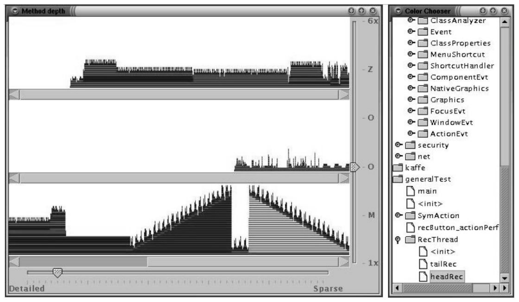
localhost:3280-www.usdoj.gov:80
Activity: Closed — Bytes Sent: 205 bytes sent
Throughput (KB/s) 0 — Received Bytes: 229 bytes received

localhost:3279-www.usdoj.gov:80
Activity: Closed — Bytes Sent: 236 bytes sent
Throughput (KB/s) 0 — Received Bytes: 224 bytes received

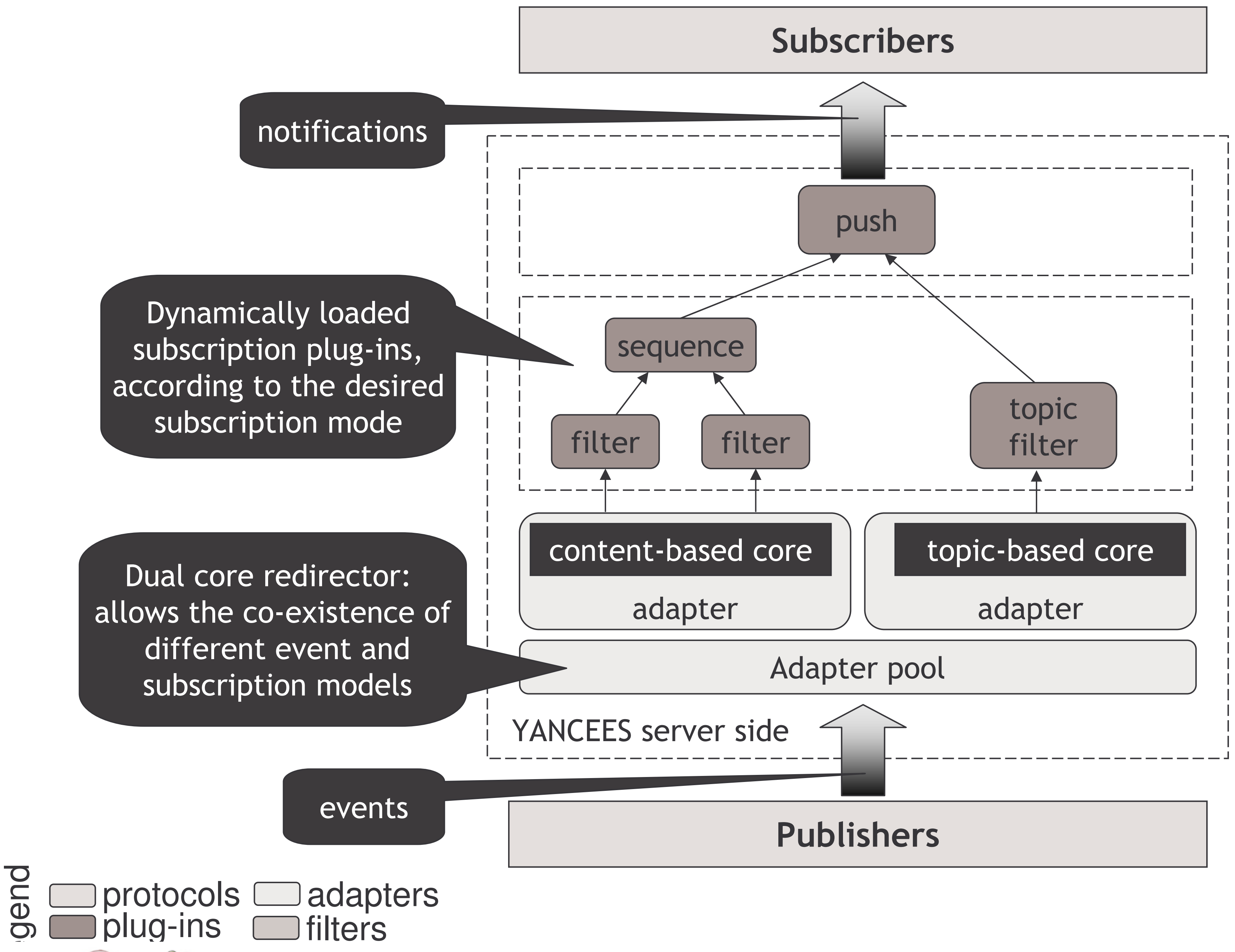
localhost:3281-www.dhs.gov:80
Activity: — Bytes Sent: 117 bytes sent
Throughput (KB/s) 0 — Received Bytes: 412 bytes received

localhost:3282-www.usdoj.gov:80
Activity: Closed — Bytes Sent: 88 bytes sent
Throughput (KB/s) 0 — Received Bytes: 339 bytes received

localhost:3283-www.usdoj.gov:80
Activity: Closed — Bytes Sent: 240 bytes sent
Throughput (KB/s) 1 — Received Bytes: 14017 bytes received

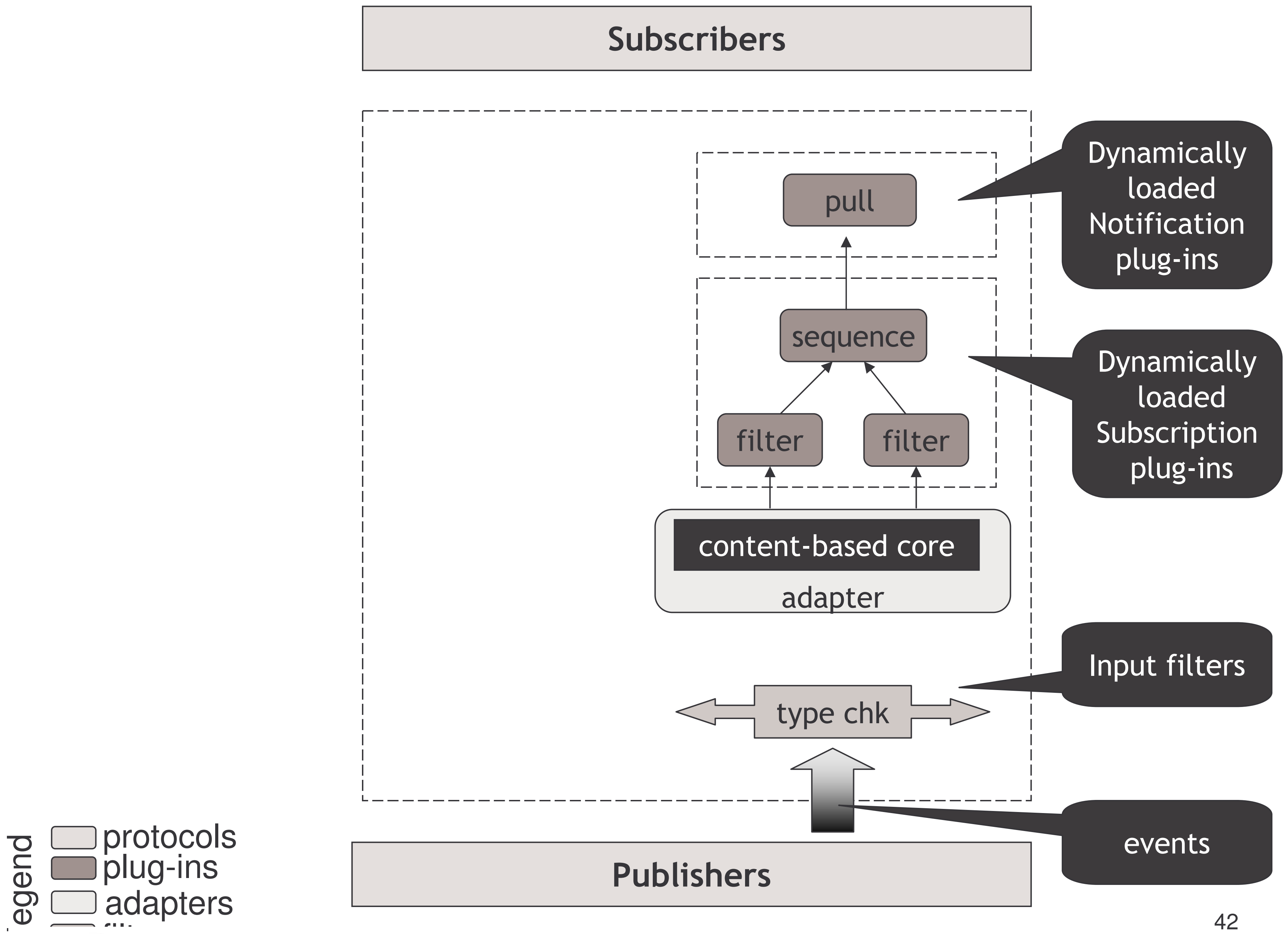


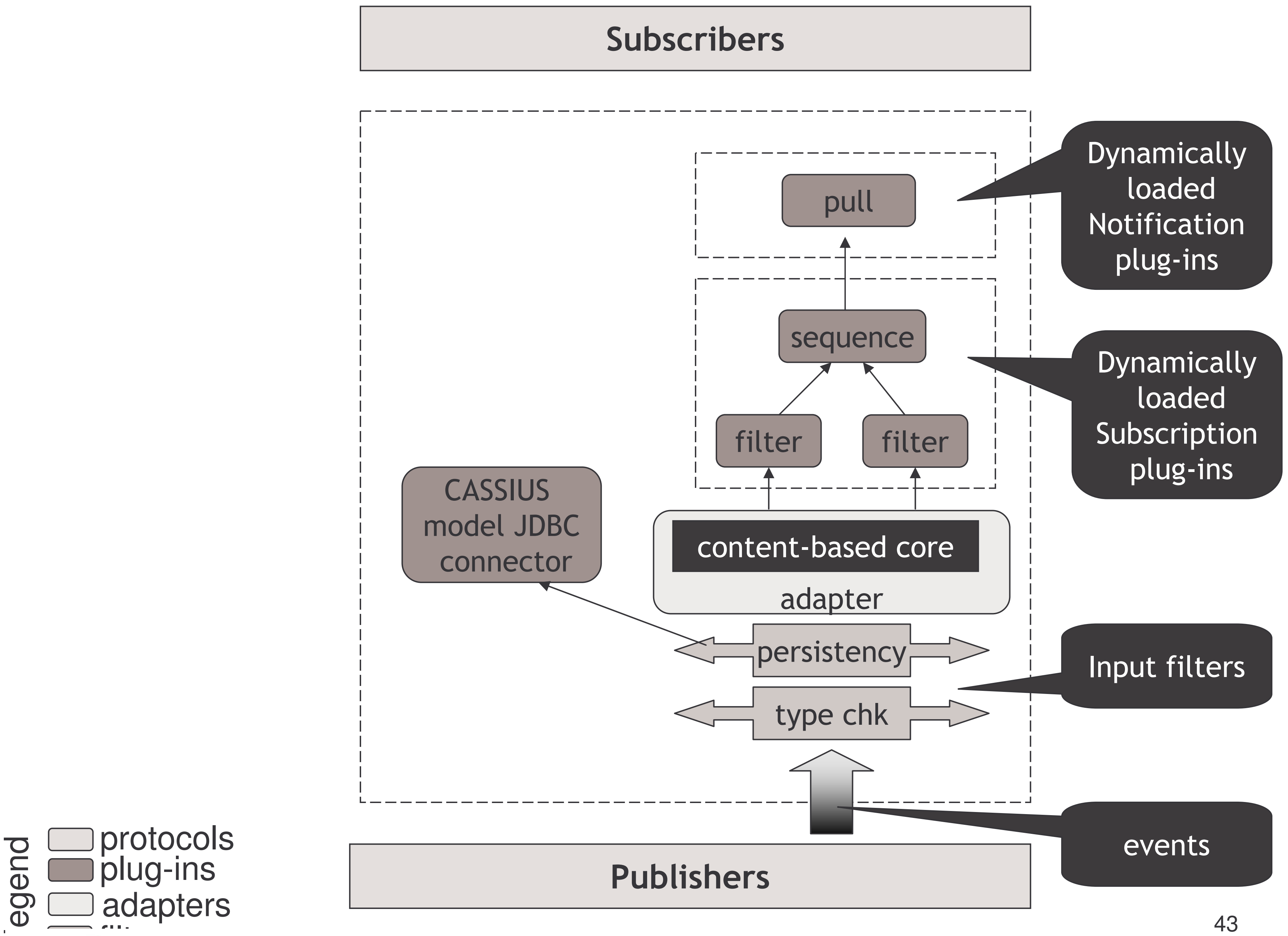
39

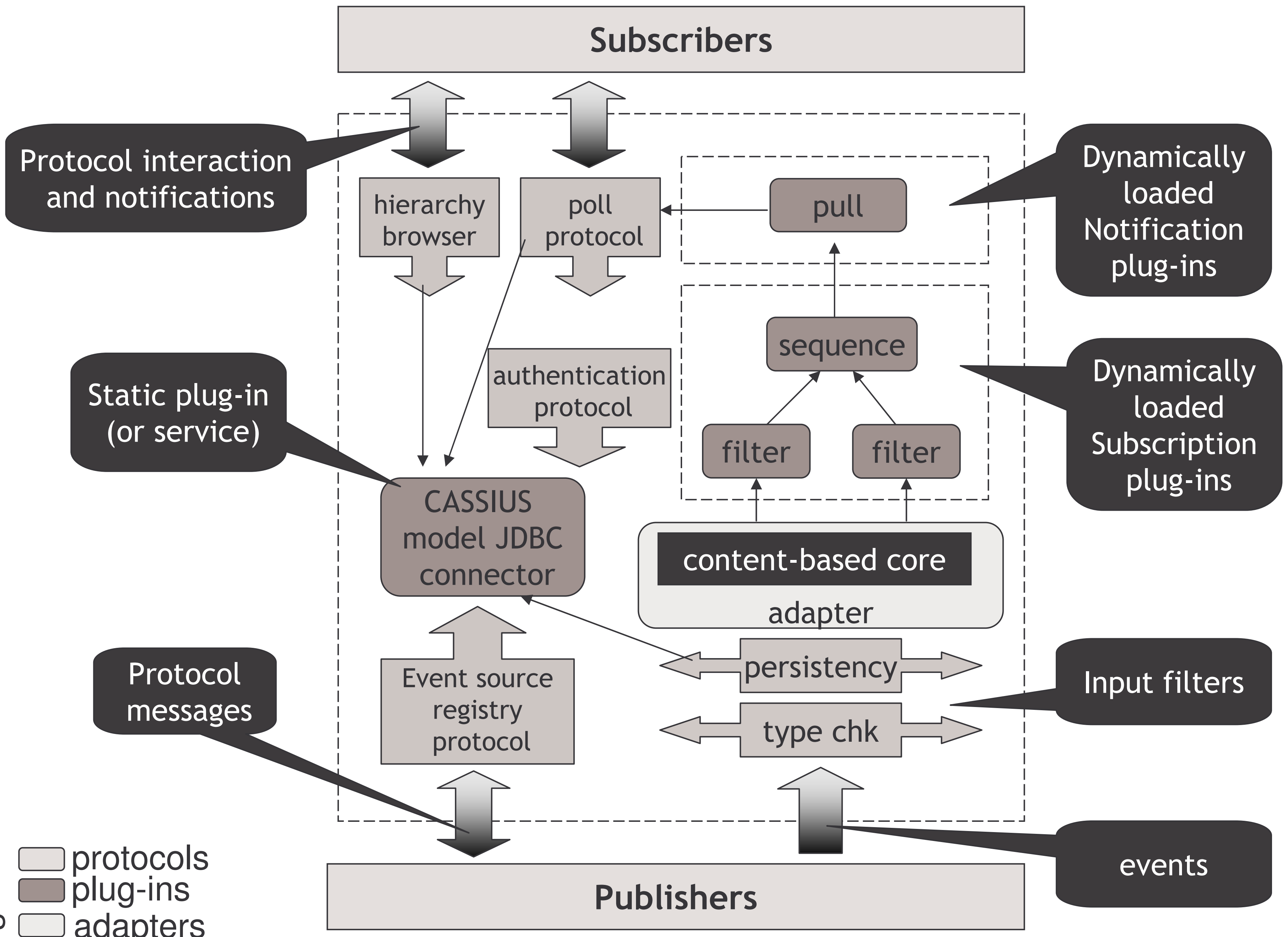

Subscribers
notifications
push
Dynamically loaded subscription plug-ins, according to the desired subscription mode
sequence
filter
filter
topic filter
content-based core
adapter
topic-based core
adapter
Dual core redirector: allows the co-existence of different event and subscription models
Adapter pool
YANCEES server side
events
Publishers
gend
protocols
plug-ins
adapters
filters

CASSIUS – awareness publish/subscribe server

- CASSIUS pub/sub model provides:
  - Event persistency
  - Event typing enforcing
  - Pull notification delivery
- CASSIUS also supports the following protocols:
  - Event source discovery
  - Event type hierarchy browsing
  - Authentication


Subscribers
pull
sequence
filter
filter
content-based core
adapter
type chk
Publishers
Dynamically loaded Notification plug-ins
Dynamically loaded Subscription plug-ins
Input filters
events
egend
protocols
plug-ins
adapters


Subscribers
pull
Dynamically loaded Notification plug-ins
sequence
filter
filter
Dynamically loaded Subscription plug-ins
CASSIUS model JDBC connector
content-based core
adapter
persistency
type chk
Input filters
events
Publishers
egend
protocols
plug-ins
adapters


Subscribers
Protocol interaction and notifications
hierarchy browser
poll protocol
pull
Dynamically loaded Notification plug-ins
sequence
filter
filter
Dynamically loaded Subscription plug-ins
authentication protocol
Static plug-in (or service)
CASSIUS model JDBC connector
content-based core
adapter
persistency
type chk
Input filters
Protocol messages
Event source registry protocol
events
Publishers

egend

▭ protocols
▨ plug-ins
▭ adapters

Conclusions

Advantages of the approach

- **Configurability**: The combination of **plug-ins and extensible languages** provide coherent composition of interdependent features;
  - the subset of language extensions and plug-ins also define the **footprint** of the server.
- **Extensibility**: new features can be provided by extending the language and implementing new plug-ins and filters
- **Reuse**: plug-ins can depend on one another, speeding up the development process
- **Support for multiple infrastructures**: the microkernel approach allows different publish/subscribe cores to be installed at the same time
- **Variability**: plug-ins can be installed at load time (configuration file) and runtime (downloaded as needed). They are also allocated according to the application needs
- **Multiple event models**: adapters to different pub/sub cores permit multiple event representations to co-exist.

Drawbacks

- Performance:
  - In our experiments, the XML technology (subscription and notification parsing) adds an extra 100 ms to the subscription process (but this is a one time cost)
  - The plug-in hierarchy adds an extra 50 ms to the notifications routing time (but the throughput is compatible with Siena and Elvin ~8000 events/second) due to our buffering strategy
- Framework costs:
  - Initial generalization and implementation
  - Initial learning curve (not much worse than more advanced pub/sub systems as CORBA-NS)
- Non-functional requirements are not so easy to implement (need to extend many points in the system, AOP may help)

Future work

- Address usability issues
  - Achieve a balance between model complexity and its extensibility
- Study the use AOP for non-functional requirements
- Study the use of rule-based patterns for more complex event processing
- Perform usability case studies

Questions/Comments?