Applications and Abstractions

A Cautionary Tale

David S. Rosenblum
Felicitous Computing Institute
School of Computing
National University of Singapore
My Net Cred

- **SIENA** Internet-scale publish/subscribe system
  - *Collaboration with Alex Wolf & Antonio Carzaniga*
- Formerly Principal Architect and CTO of
- Confidentiality in Internet-scale publish/subscribe
- **ROAR**: Rendezvous on a Ring
  - *PhD of Costin Raiciu, collaboration with Mark Handley*
- Some papers in ACM TOCS, PODC, SIGCOMM, ICNP
- Ten patents for work at
Question 0

What is (an) abstraction?

“the process of considering something independently of its associations, attributes, or concrete accompaniments”

[Oxford American Dictionary]

- Implementation independence
- Widespread applicability and reusability
Question 1
Why are abstractions needed?

• for understanding and reasoning
• for designing and implementing

My focus in this talk is on abstractions for building applications that are to be deployed on the Internet
Question 2
What abstractions are needed?

- Communication paradigms
- Storage paradigms
- Structuring and coordination paradigms
- Formal logical models of these
- Formal quantitative models of these

My own interests are in communication paradigms and probabilistic models
General-purpose abstractions for building applications can lose their generality and/or abstractness once realized at Internet scale.

There may be many approaches for realizing an abstraction, but each one employs its own assumptions, algorithms, protocols, optimizations and heuristics.

Those choices can strongly constrain the set of applications able to use the realization naturally, effectively and efficiently.
Motivating Example

Publish/Subscribe

- Natural abstraction for multi-way, asynchronous dissemination of data
- At application level, middleware or brokers provide decoupling, anonymity, matching, caching, authentication, and many other services
- Many conceivable applications at Internet scale

---

<table>
<thead>
<tr>
<th>Applications</th>
<th>Components</th>
<th>Objects</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>notifications, alerts, updates</strong></td>
<td><strong>events</strong></td>
<td><strong>events</strong></td>
<td><strong>signals, interrupts</strong></td>
</tr>
</tbody>
</table>
Internet-Scale Pub/Sub Applications

symbol = “AAPL” and price > 700.00

symbol = “AAPL”, price = 701.23, shares = 5000, [etc.]

Stock Quotes
Internet-Scale Pub/Sub Applications

$\text{bus} = (10 \text{ or } 30 \text{ or } 51 \text{ or } 143 \text{ or } 188) \text{ and } \text{nextnextstop} = 16069$

$\text{bus} = 143, \text{capacity} = 0.9, \text{stop} = 16089, \text{nextstop} = 16079, \text{nextnextstop}=16069$

Location-Dependent Travel Alerts

$\text{bus arrivals, taxi dispatching, traffic incidents, etc.}$
## Internet-Scale Pub/Sub Application Characteristics

<table>
<thead>
<tr>
<th>Subscriptions</th>
<th>Application</th>
<th>Notifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selectivity</td>
<td>Stock Quotes</td>
<td>Frequency</td>
</tr>
<tr>
<td>Churn</td>
<td>Software Updates</td>
<td>Uniqueness</td>
</tr>
<tr>
<td></td>
<td>Travel Alerts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>News Alerts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MMOGs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Battlefield Awareness</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Location Updates</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Social Network Alerts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Context Awareness</td>
<td></td>
</tr>
</tbody>
</table>

- **Selectivity**: Indicates the number of subscribers interested in a particular application.
- **Churn**: Refers to the rate at which subscribers are added or removed from an application.
- **Stock Quotes**: High selectivity and churn.
- **Software Updates**: Low selectivity and high churn.
- **Travel Alerts**: High selectivity and low churn.
- **News Alerts**: High selectivity and high churn.
- **MMOGs**: Very high selectivity and very high churn.
- **Battlefield Awareness**: High selectivity and high churn.
- **Location Updates**: Low selectivity and high churn.
- **Social Network Alerts**: High selectivity and low churn.
- **Context Awareness**: Very high selectivity and very low churn.
SIENA

- General-purpose realization of publish/subscribe at Internet scale
- Designed as a decentralized overlay of brokers
- Full content-based matching of notifications to subscriptions with best-effort delivery
- Self-describing notifications—no notification types, predefined topic hierarchies, etc.
SIENA

Subscription Forwarding

s1: “price < 700”
SIENA

Subscription Merging

s1 covers s2

s2: “price < 600”
SIENA
Notification Delivery

n1: “price = 550”
SIENA

Implied Ideal Application Characteristics

• Many publishers and many subscribers
  To justify decentralized implementation

• Notifications much more frequent than subscriptions
  To justify subscription forwarding

• Low subscription churn
  To justify subscription forwarding and merging

• High subscription selectivity
  To justify content-based matching in brokers

• Subscription similarity correlated with network locality
  To justify subscription merging
SIENA

Implied Ideal Application Characteristics

- Many publishers *and* many subscribers
  *not Stock Quotes*

- Notifications much more frequent than subscriptions
  *not Software Updates*

- Low subscription churn
  *not location-dependent applications*

- High subscription selectivity
  *not Software Updates*

- Subscription similarity correlated with network locality
  *not Stock Quotes, Software Updates, MMOGs, etc.*
Few applications have all these characteristics

- Traffic alerts
- Social interaction alerts
- Others?
Other approaches induce similar limitations

- Gryphon
  - Subscription flooding over tree of clusters
  - *Applicable if subscriptions are few and stable*

- Hermes
  - Rendezvous nodes allocated to content types
  - *Applicable if load is spread evenly by type*

- PreCache
  - Trie- and *kd*-tree-based subscription storage
  - *Applicable if subscription churn is very low*
Conclusion

• Conceptually, publish/subscribe is a very general abstraction

• But it loses generality once realized at Internet scale

• And it does so for reasons that have little to do with the peculiarities of the Internet

• Adaptability as a compromise
  - ROAR’s partitioning/replication tradeoff
  - Alex and Antonio’s content-based networking (CBN)
Question 3
How can research ... be fostered ... ?

• With respect to abstractions for building ...
  I would like to have better formal logical and probabilistic models ...
  ... for exploration of and reasoning about ...
  ... the design space induced by a network abstraction like publish/subscribe.