

# “Towards Next Generation Pervasive Internet”

Dag Johansen

Department of Computer Science  
University of Tromsø  
Norway





# Outline

---

1. Internet problems revised.
2. Towards a proactive Internet.
3. An approach: WAIF.



# 1. Internet Problems Revised

---

- Trustworthy computing – Internet dependable infrastructure.
  - Security:
    - ☒ Tampering: unauthorized alteration of information.
    - ☒ Leakage: unauthorized recipient of information.
    - ☒ Vandalism: interference with proper operation.
    - ☒ .....
  - Fault-tolerance:
    - ☒ Availability.
    - ☒ Reliability.



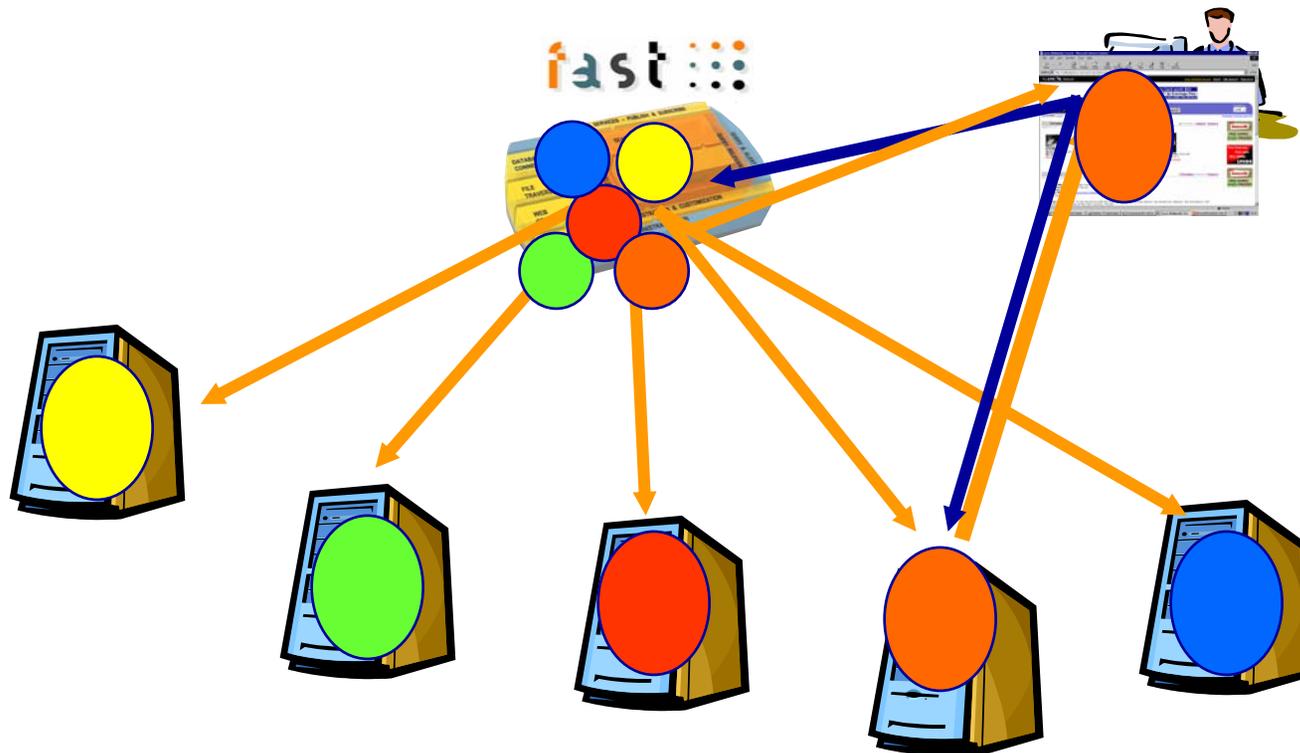
# 1. Problems Revised (cont)

---

- 600.000.000 users:  
→ *scalability and performance.*
- 3.000.000.000+ web pages:  
→ *Recall and precision:*
  - ☒ “Recall” (rcvd relevant data/all relevant data = ~1).
  - ☒ “Precision” (rcvd relevant data/all data rcvd = ~1).

# 1. Problems Revised (cont)

”Mediators” – cope with scale and complexity.

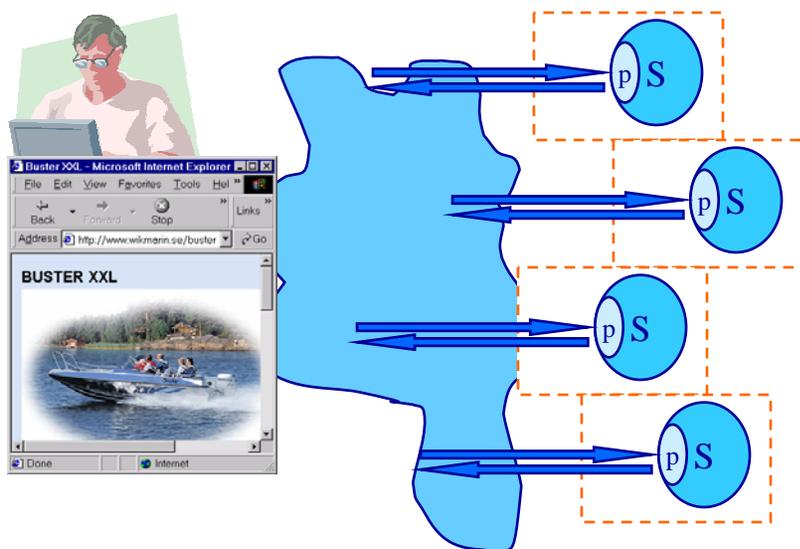


Google: 76,800,000 hits for Microsoft.

84,700 for Holmenkollen and 84,200 for Gary Moulton.

# 1. Problems Revised (cont)

- *Inherited structure* from the initial Internet: **client-server**.



- Fact: interaction model that takes (user) time.



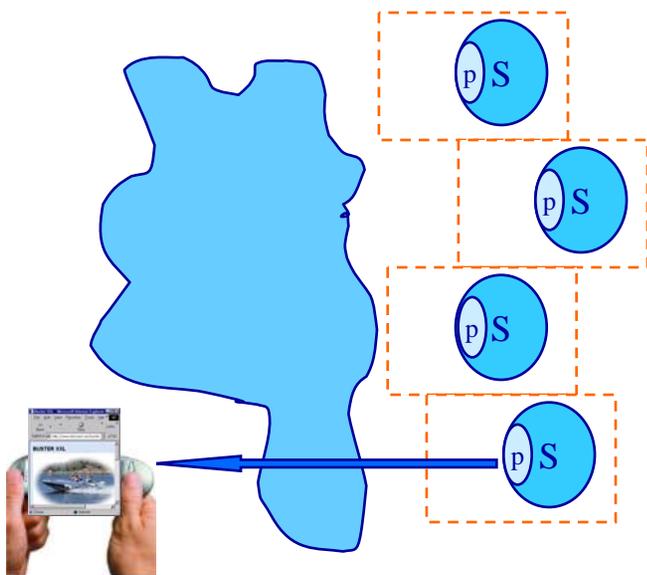
## 2. Towards a Proactive Internet

---

- We conjecture that the Web's next paradigm shift will include a much more proactive computing model.
- This will transform a passive web being searched by users, to information and service providers searching actively for users.

# Proactive Internet

- The web works autonomously on your behalf and notifies you.



## Goals:

- High recall.
- Extreme precision.
- Context-aware.
- Real-time.



# Evolution

---

... important demands, structures and building blocks of a proactive Internet are already getting there.



## (a) Users – regular consumers

---

- Average computer users are regular consumers, no computer experts.

*→ Users want appropriate UI paradigms with same simplicity and transparency as with electricity, tap water, television, and cars.*

## (b) User profiling

---

- Users leave explicit and implicit traces on the Internet (Passport data, orkut data, service registration data, credit card numbers, spyware, cookies, logs, .....

*→ User behaviour and personal information continuously captured and stored.*

## (c) Subscription services

---

- Subscription based services emerging (headlines from news sources, RSS feeds, stock alerts, direct marketing, recommendations .....

→ *Users want push information (but not spam).*



## (d) Computers have changed

---

- Interactive computers are not dominating  
(“2% of the processors shipped in 2000 for interactive computers”,  
D. Tennenhouse, Intel, in Comm. ACM, May 2000).  
  
→ Embedded computers:
  - *Contact with the physical environment, not humans.*
  - *Tie physical to electronic devices.*
  - *Speech or sensing driven, not keyboards.*
  - *Reactive devices, monitor and control.*
  - *Aggregation services (sensor networks).*

## (e) Many computers

---

- Ubiquitous or pervasive computing; users surrounded by computers:
  - Cellulares and PDAs, home and office computers.
  - Integrated in electronic devices, clothing, clocks, walls, cars, ...

*→ Past the one-computer-one-person breakpoint.*



## (f) Connectivity

---

- Wireless, inexpensive networks emerge.

→ *Mobile users by default network connected.  
..... but not "enough-speed bandwidth" yet.*



## (g) Remote computing commodity

---

- Average users participate in *peer-to-peer* file and cpu sharing.

→ *Remote computers can be used as a personal (GRID) computing platform.*

## (h) Run-time extensibility

---

- Modular software and platforms:
  - Web services (SOAP, XML, WSDL, UDDI).
  - Platforms and programming languages (.NET, Webspheres, Java, JXTA, Jini, .....

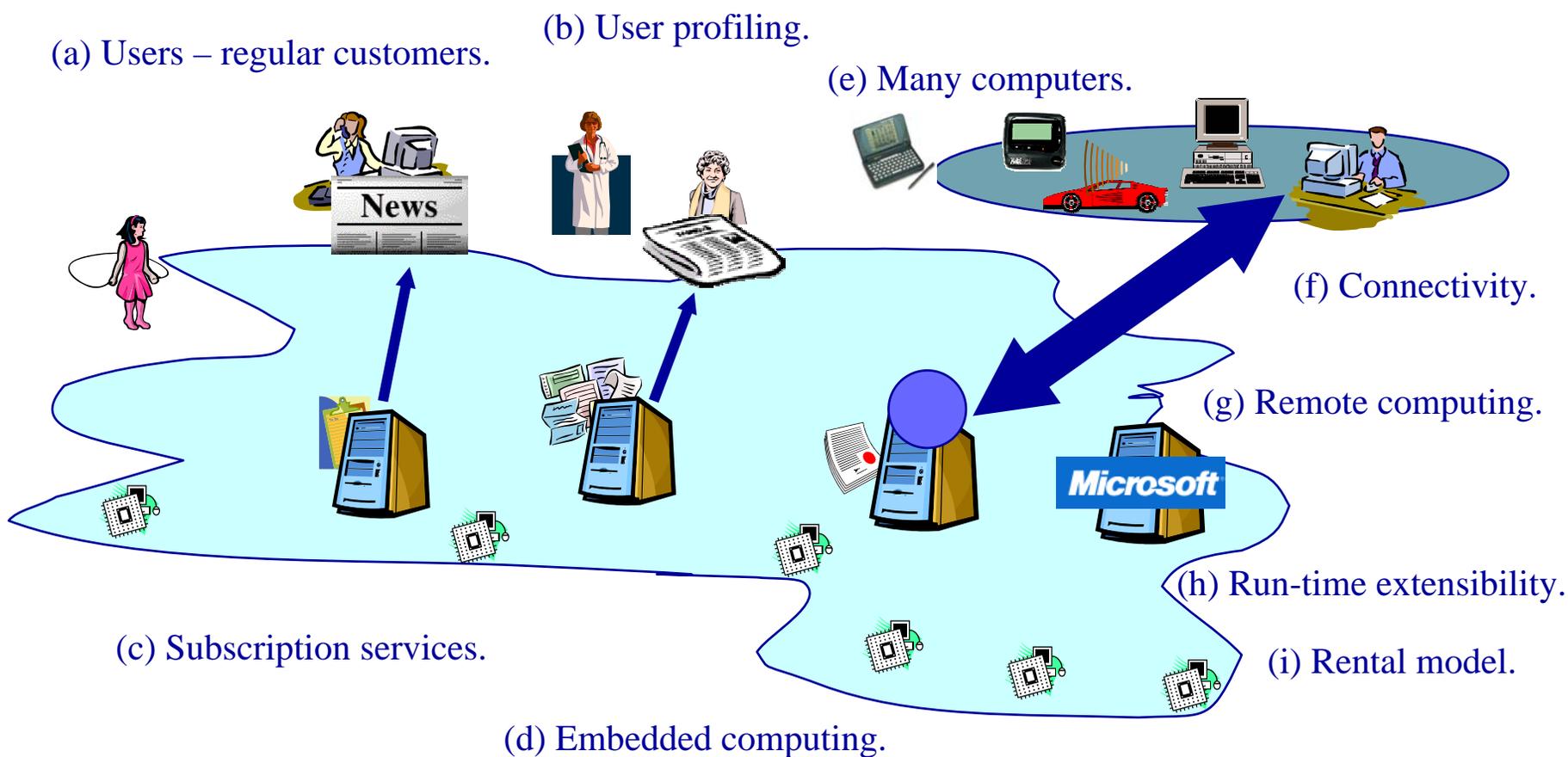
→ *One-time server programmability not a problem anymore.*

## (i). Rental model

---

- "Pay by use" software concept emerging.
  - *Users do not license all software, they configure (and pay for) applications and services on demand.*

# This gives us ... .. complexity





## 3. WAIF (Wide Area Information Filtering)

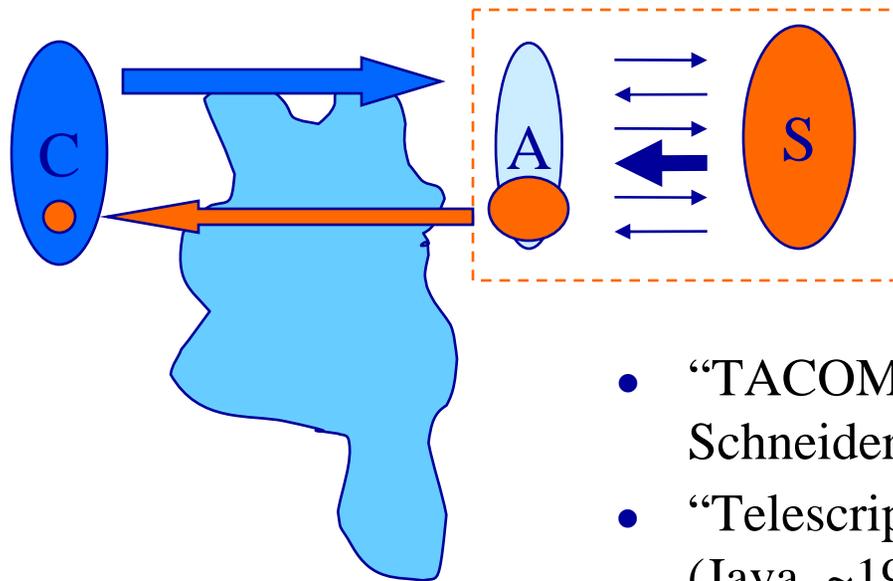
---

- Problem: *"How to structure the next generation web."*
- Approaches:
  1. Use the network as a *personal* computer:
    - **move personalized computational tasks around.**
  2. *Proactive computing* combined with high precision:
    - **humans not in the loop, but above the loop.**
  3. Mobile users in a *pervasive computing* environment:
    - **design for mobility.**
- International cooperation: University of Tromsø, Cornell University, and UC San Diego.

***"Get Rid of the Computer!"***

# Mobile agent technology

- Mobile code: program and install autonomous code (A: Python, C, Perl, Tcl, Java, Scheme) and data at remote servers.

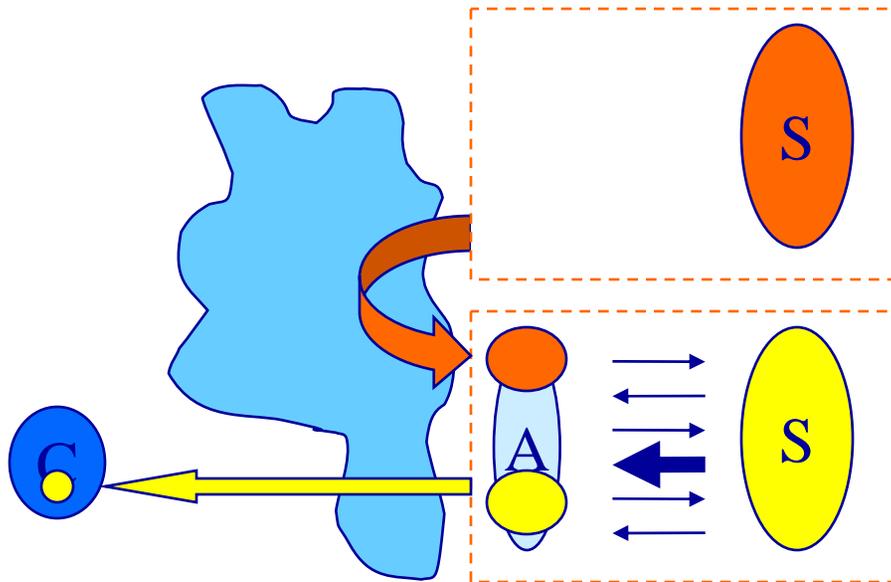


- “TACOMA”; Johansen, van Renesse & Schneider; 1994.
- “Telescript” & General Magic, White; 1994. (Java, ~1996).
- Jim Miller, Microsoft: “in 1997 build a mobile agent system supporting multiple languages”  
→ .NET CRE.

# Technology that failed?

---

- Itinerant style computations (100+ groups, most Java).

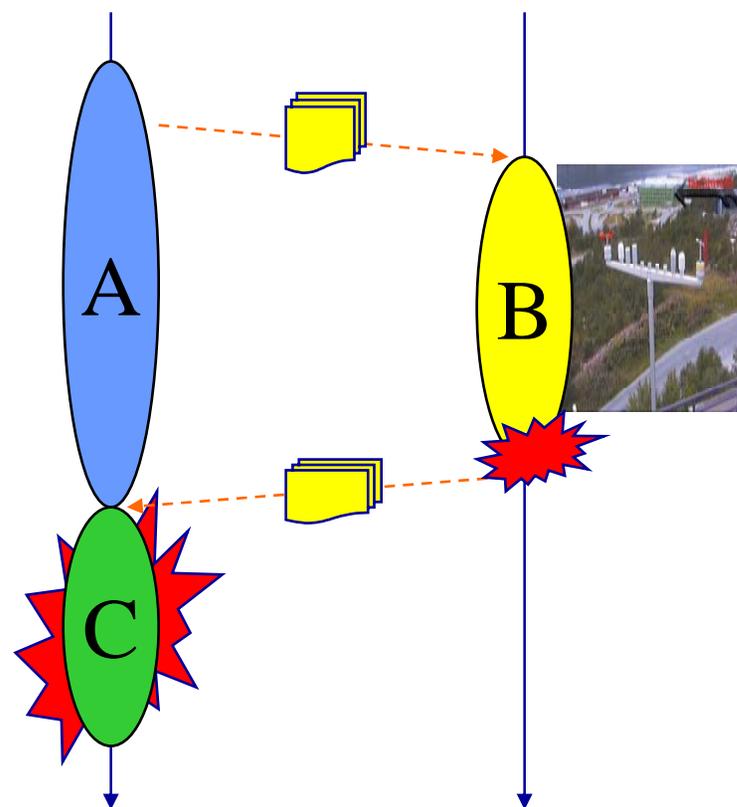
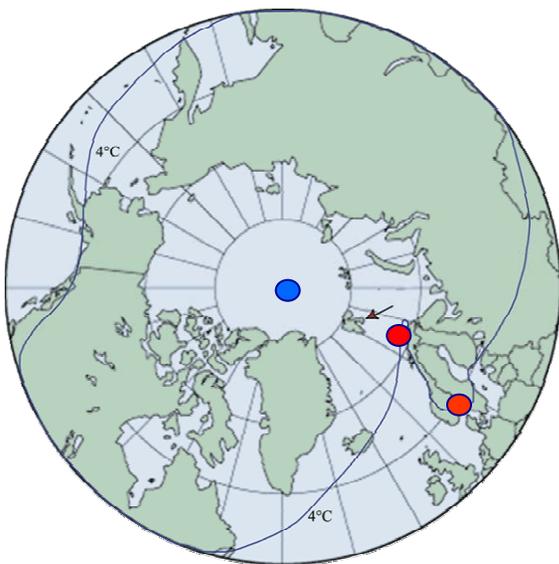


- Problems: security, infrastructure, “killer application”, .....

## “StormCast Alarms”

- Problem 1993: extend weather and sensor networks at run-time (Coast guard, trawlers, Pasvik, .....

Example: ‘temp > 20 & wind < 10’



## Operational (1997)

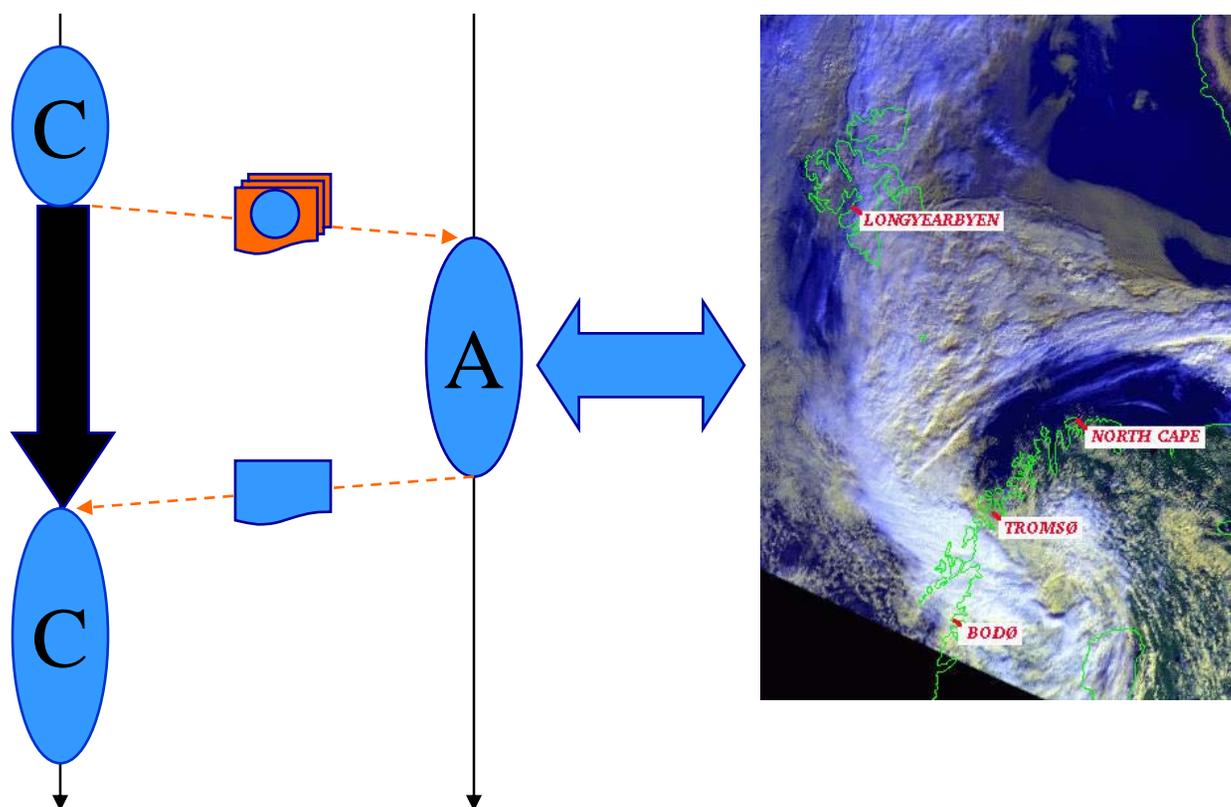
---

- Installs using browser, PDA, or cellular.
- Notifies through e-mail, browser, or cellular.
- Scales: 1000 alarms run in 823 millisecc.

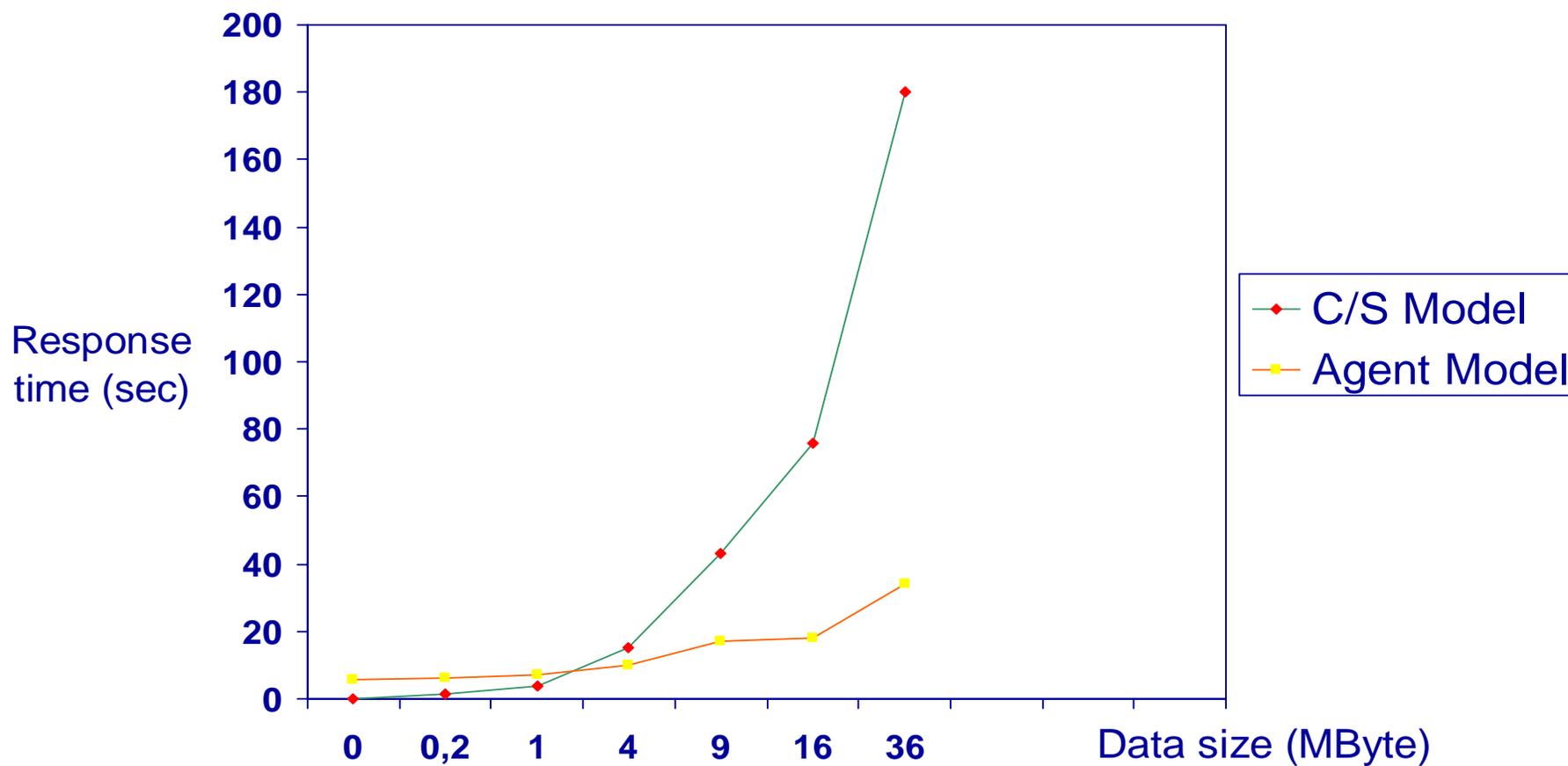


## “TACOMA Image Server” (1998)

- Tromsø Satellite Service: Terabyte datastore with satellite data.
- Ship customized code (nK) to the data source (SQLCLR, Yukon).



# Varying Data Size



HP 720, TACOMA v.1.2, 10 Mbit/s Ethernet.

## Lessons Learned

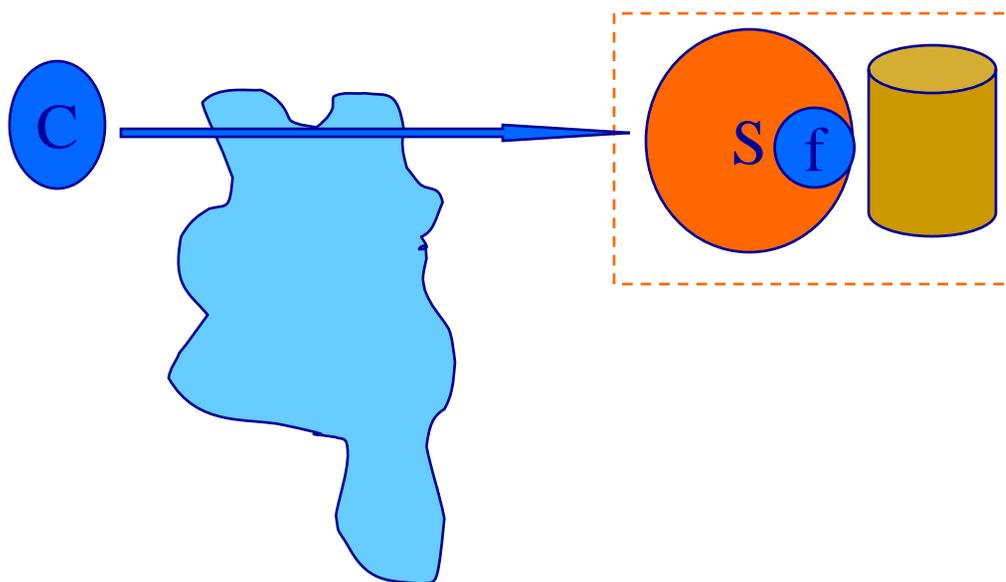
---

- Mobile code is convenient for run-time software installation:  
→ extensible servers.
- Single-hop agents the normal case, multi-hop the special case.
- Languages orthogonal to the system: Multi-language support important.
- Agent integrity hard.

# WAIF-based Internet

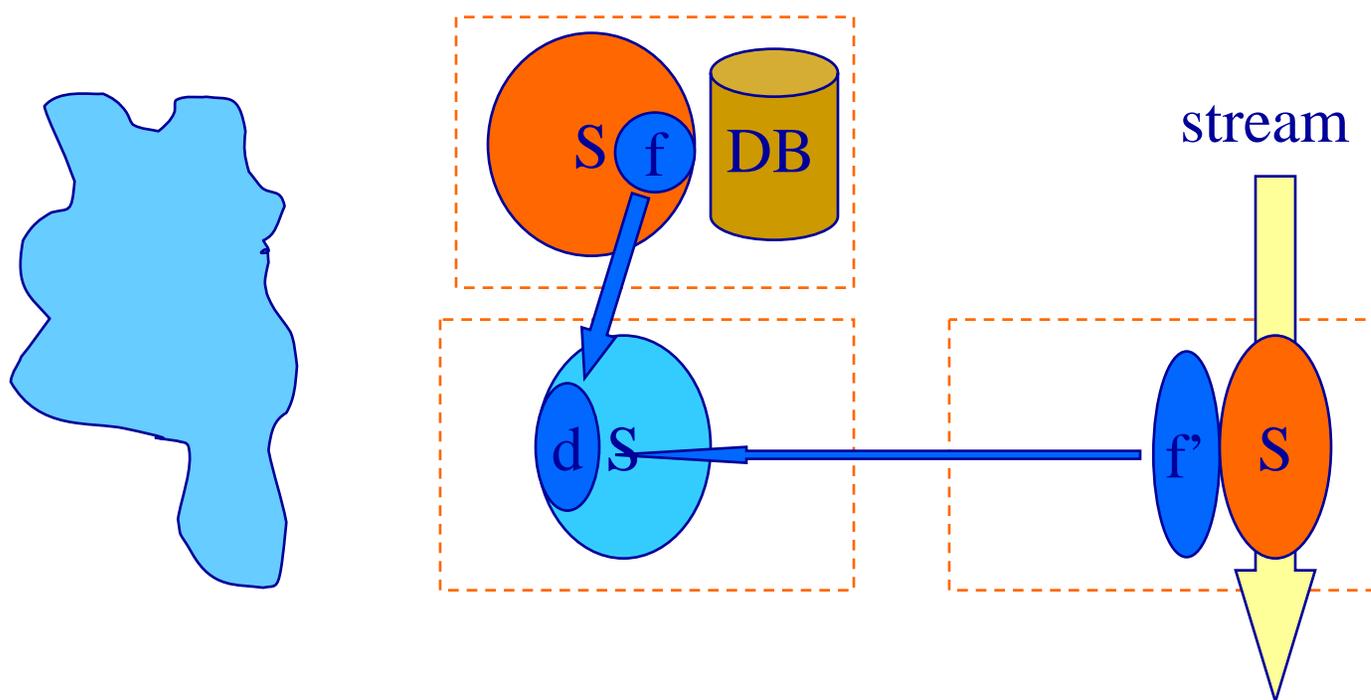
---

- Mobile agent lessons: Install software components remotely.



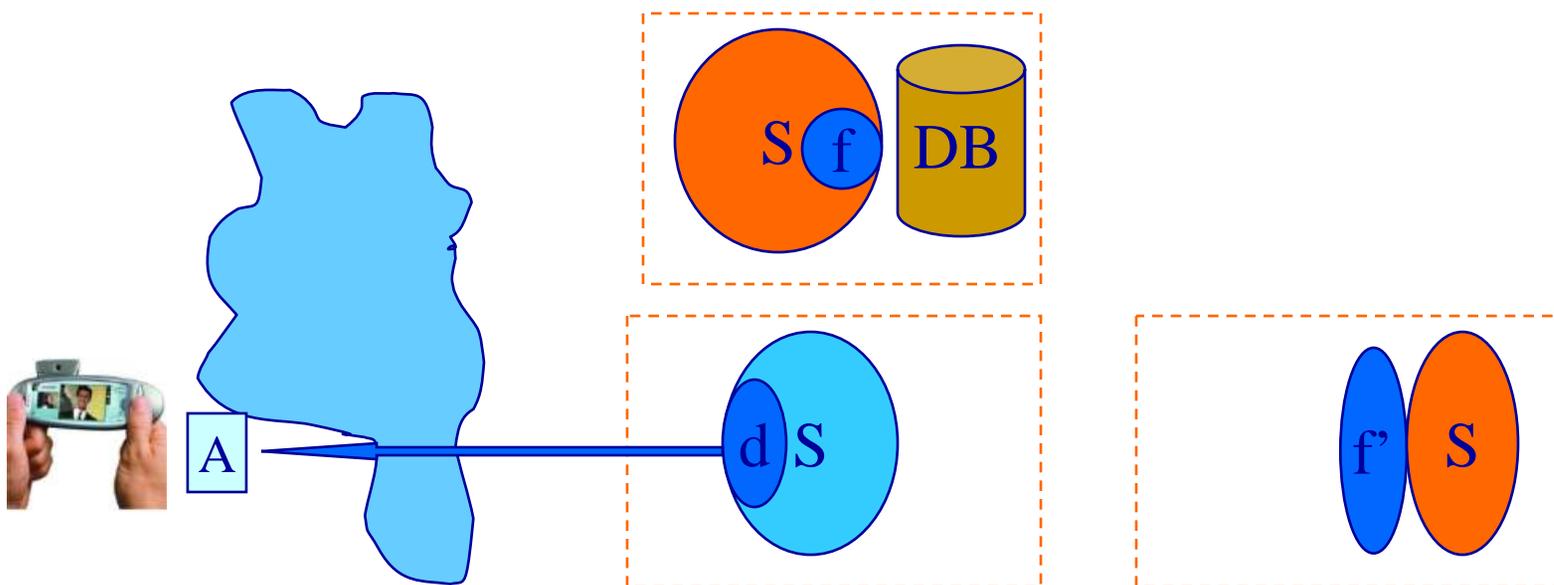
# WAIF-based Internet

- Autonomous computing in the network.



# WAIF-based Internet

- High-precision alerts.





## Example Alert

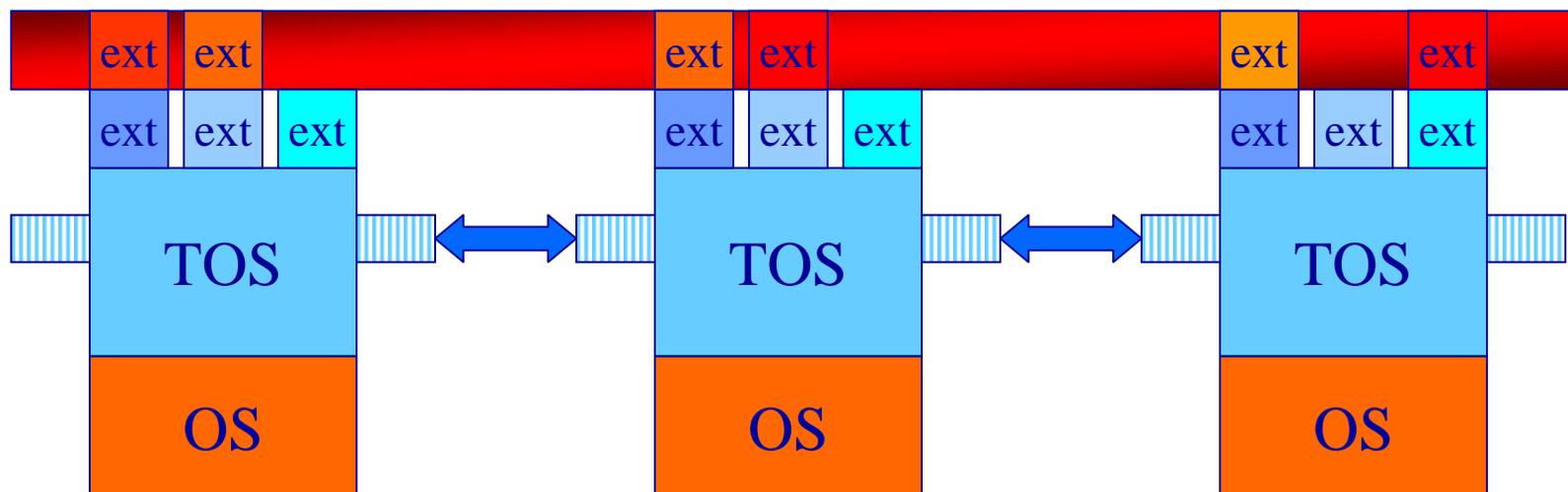
---

```
event = self.in.get()
if event[´topic´] == ´sport´:
    self.out.push(event)
```

```
#upon msg: mk event
#check event type
#send sport event
```

## ”WAIF Run Time Environment”

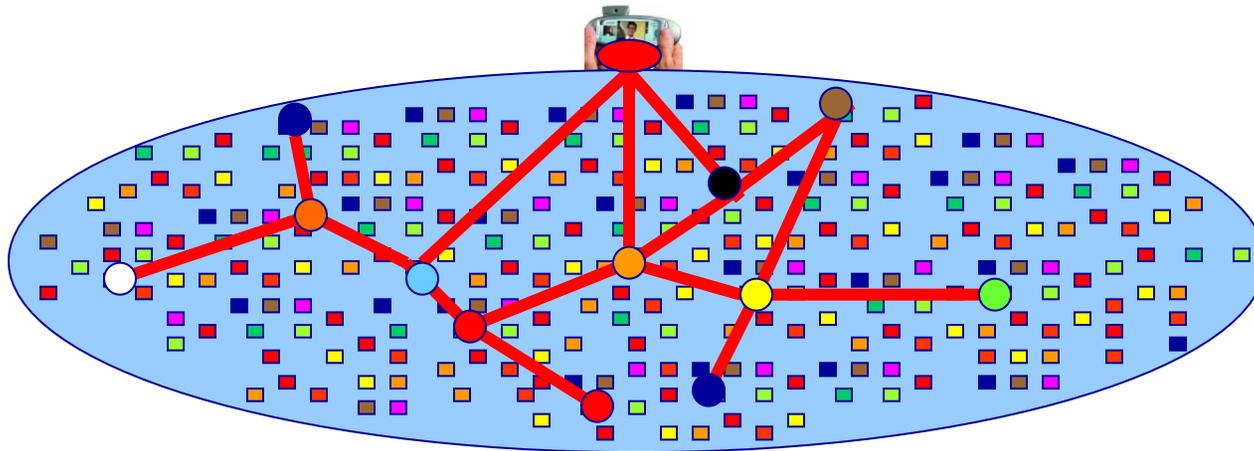
- WAIF kernel at each remote server:
  - TOS: server side WAIF run-time (Håvard Johansen).
  - Vortex kernel: Scalable performance isolation (Åge Kvalnes).
- Dynamic configuration of the WAIF servers.
- **Install personalized application components.**



# "Personal Overlay Network System"

---

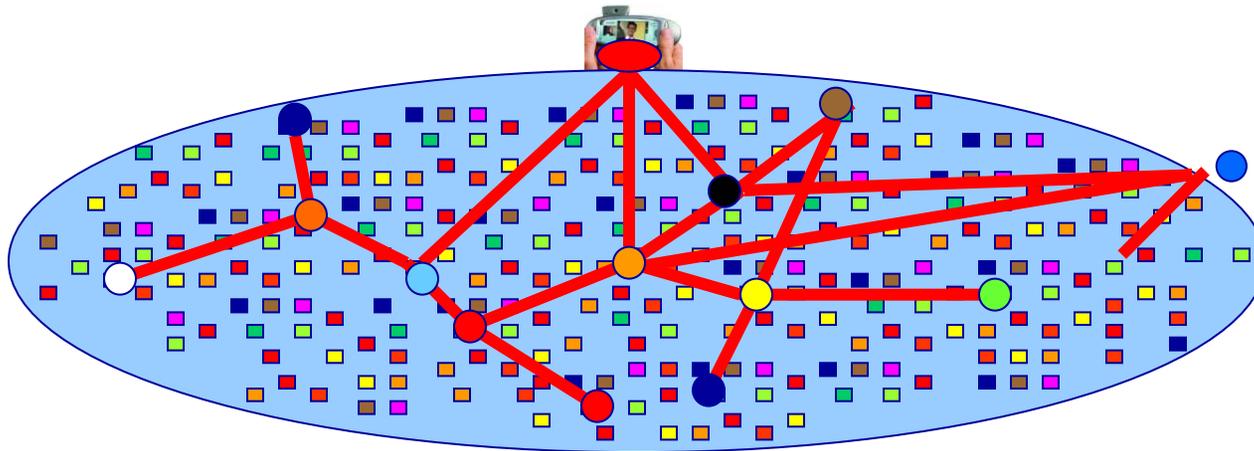
- Locate services and create your personal distributed system as an overlay network (*PONS*).



# Supports Mobile Users

---

- Move *user environments* transparently along.



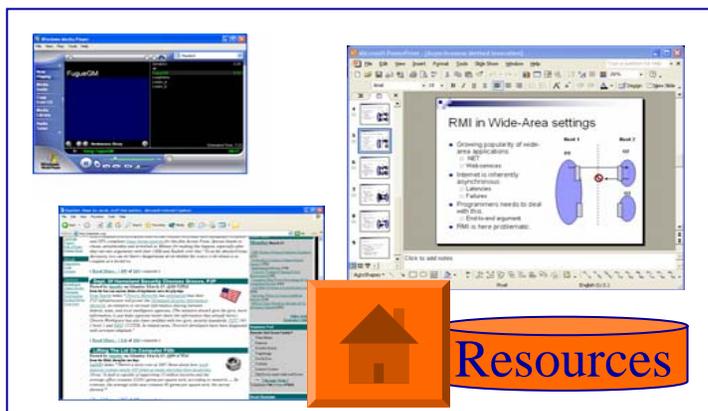
- Approach: Move *task description*, meta data and application data.
- Alternative: migrate or *lazy-copy* application code.

# Task Migration

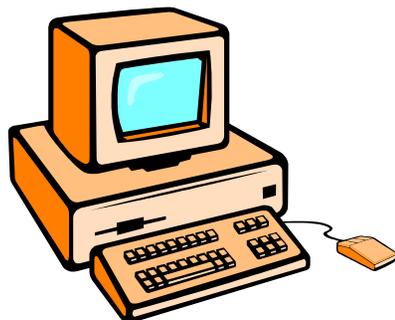
---

- Legacy applications (Word, IE, ppt, mp3, games ...).
- Playing with different transport mechanisms:
  - $PC^1 \rightarrow \text{USB stick} \rightarrow PC^2$ .
  - $PC^1 \rightarrow \text{network file system} \rightarrow PC^2$ .
  - $PC^1 \rightarrow \text{mobile agent system} \rightarrow PC^2$ .
  - $PC^1 \rightarrow \text{central server} \rightarrow PC^2$ .
  - $PC^1 \rightarrow \text{e-mail} \rightarrow PC^2$ .

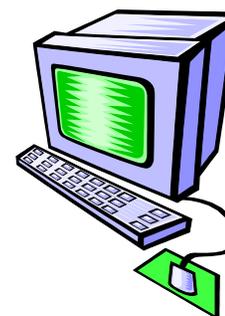
# Everyday Problem



Work



Home

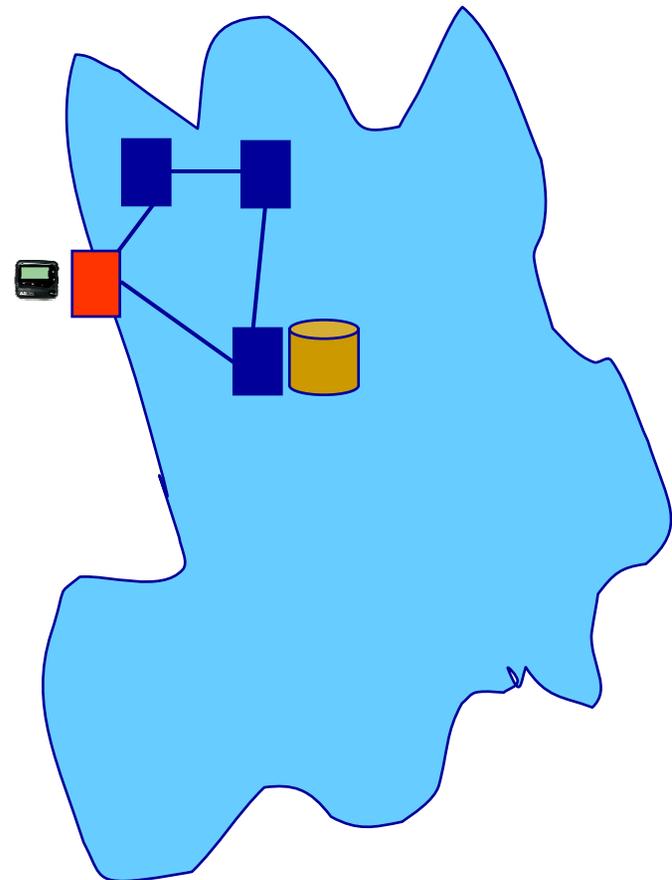


Demo next break: Are Johannessen, Kai-Marius Pedersen, and Andre Søreng.

# Software Architecture

---

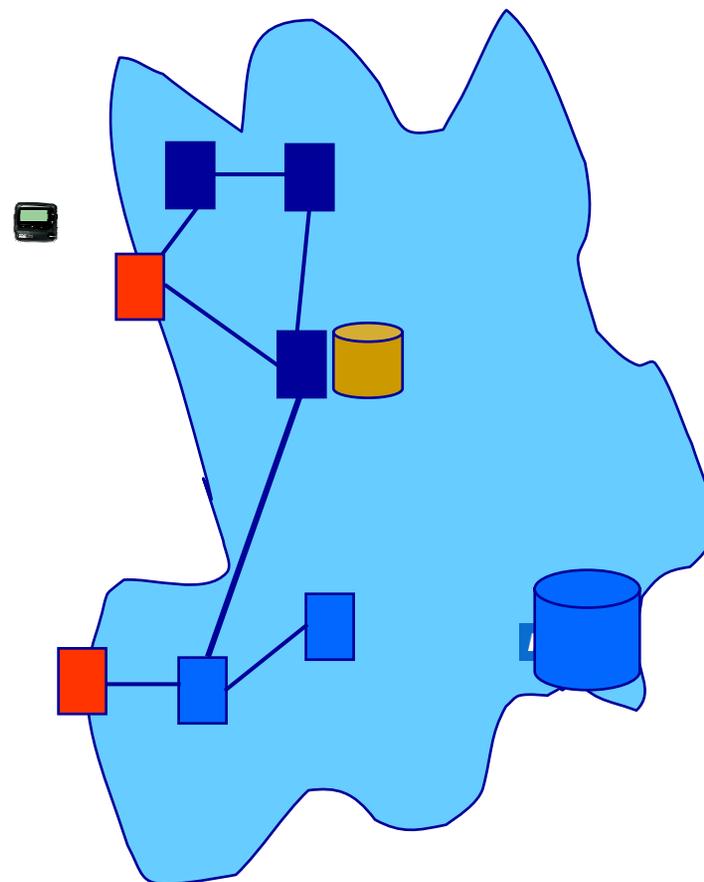
- Pervasive computing: environment saturated seamlessly with computers, sensors and communication facilities.



# Software Architecture

---

- Pervasive computing: environment saturated seamlessly with computers, sensors and communication facilities.
- Run-time configuration renting services from the environment (3<sup>rd</sup> parties).





## Related work

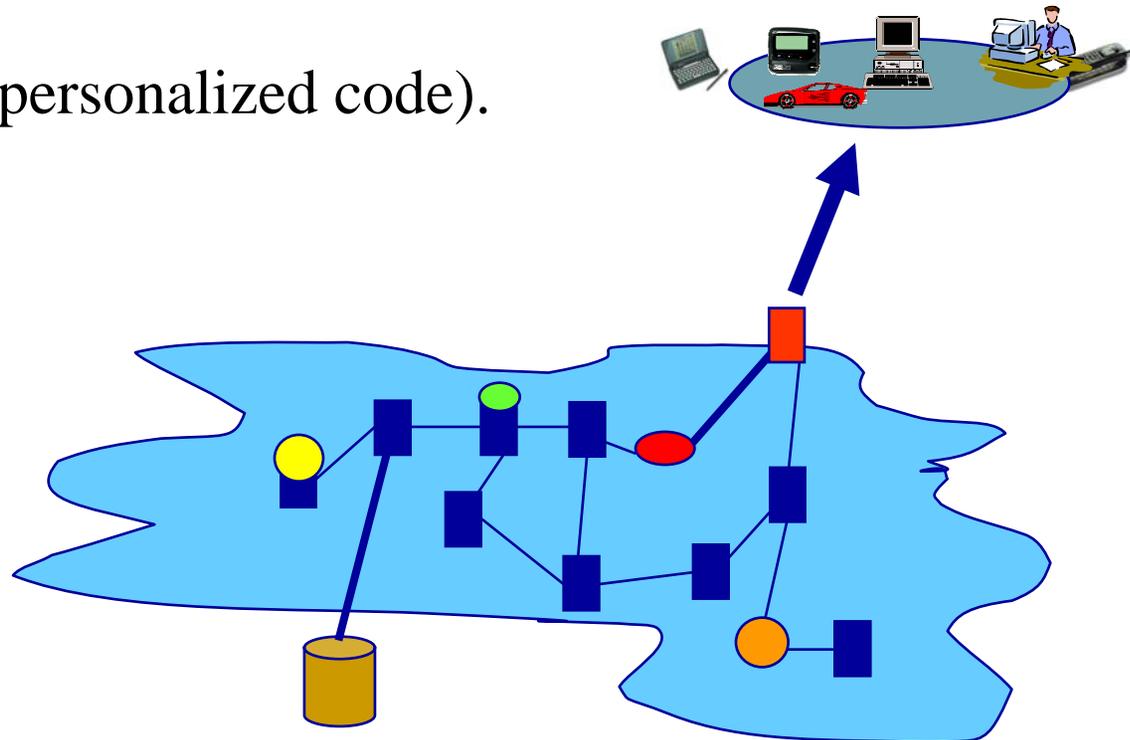
---

- Web services (Microsoft, IBM, BEA).
- Haystack and Oxygen (MIT).
- Oceanstore (Berkeley).
- Spinglass (Cornell).
- Semantic Web (W3C).
- Autonomic Computing Initiative (IBM).
- Pervasive computing, Pastry (DHT), Scribe (Microsoft Research).

# Summary

---

- Next generation Internet:
  - Pervasive.
  - Extensible (personalized code).
  - Push based.





<http://www.waif.cs.uit.no>

---

Questions?