Internetware

A Software Paradigm for Internet as a Computer

Gang HUANG
Peking University
2011-03-23, Beijing

Agenda

- Internet As A Computer
- Software of this NEW Computer
- Challenges Addressed by Internetware
- Internetware Research and Practice

Internet as a Computer (Internet Computer)

- **Internet is evolving to a Global Ubiquitous Computer**
  - **Many big and hot trends in IT research and business try to study such evolution from different perspectives**

Technical Trend

- Semantic Web
- Social Computing
- Service Computing
- System of Systems
- Pervasive Computing
- Grid/Cloud Computing
- Internet of Things

Big Trend

Business Trend

- Digital Economy
- E-government
- Internet Culture
- Social Network
- Modern Service
- Virtual World
- Smarter Planet

•Grid/Cloud computing proposes a new model of networked applications from the perspective of resource sharing and management.
•Pervasive computing discusses a new situation of networked applications from the perspective of human computer interaction.
•Service Oriented Computing focuses on a new form of software with emphasis on collaboration and dynamism from the philosophy of software as a service.
•…

IBM Microsoft Google amazon salesforce.com

Scenarios of Internet Computer

Cooperation On Demand (emergent)

- more and more goals are unclear, unexpected or un-deterministic before the cooperation finally happens (e.g. real-world emergency)
- cooperators may have no or little relations before and after

Airport Management

Passenger List and Info

Thermometers

Chinese Center for Disease Control and Prevention

2. CDC acquires all customers in the same flight

3. CDC notifies all customers go to the given hotels

1) Contact CDC when suspected H1N1 infectors are detected

Hotel

4. CDC sends customer info to the hotels

Hospital

5. CDC notifies the hospitals go to the hotels and take case of the customers

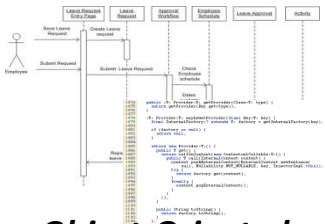
H1N1 Control and Prevention

Software of Internet Computer

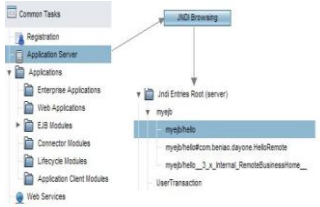
"Internet Computer" requires substantial improvements in software characteristics for implementing new business naturally with new technology.

- Before "Internet as a Computer", the goal of cooperation is predefined while the partners and interactions can be fixed or not
- Such emergent co-operations are uneasy, labor-based and un-trusted today

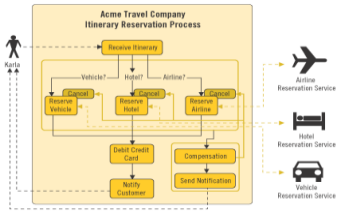
Hardwired (fixed partner, fixed interaction) → Loosely-coupled (unfixed partners, fixed interaction) → Flexible (fixed partners, unfixed interaction) → Goal-driven (unfixed partners, unfixed interaction) → Emergent (induced goal)



Object-Oriented

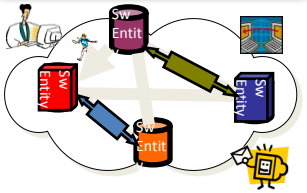


jee: jndi-lookup



Service Flow

Dynamic service flow



Labor-based emergent cooperations can be observed in such as mashups and end user programming

Internet Computer Software Characteristics

Cooperative: software can interact with others in static, dynamic and even on demand manners

Emergent: software may have un-designed behaviors or un-expected effects on its runtime instances or interactions with others

Situational: software is capable of perceiving its runtime context and scenarios

Evolvable: software is easy to add, remove and change its functionalities on-the-fly and just-in-time

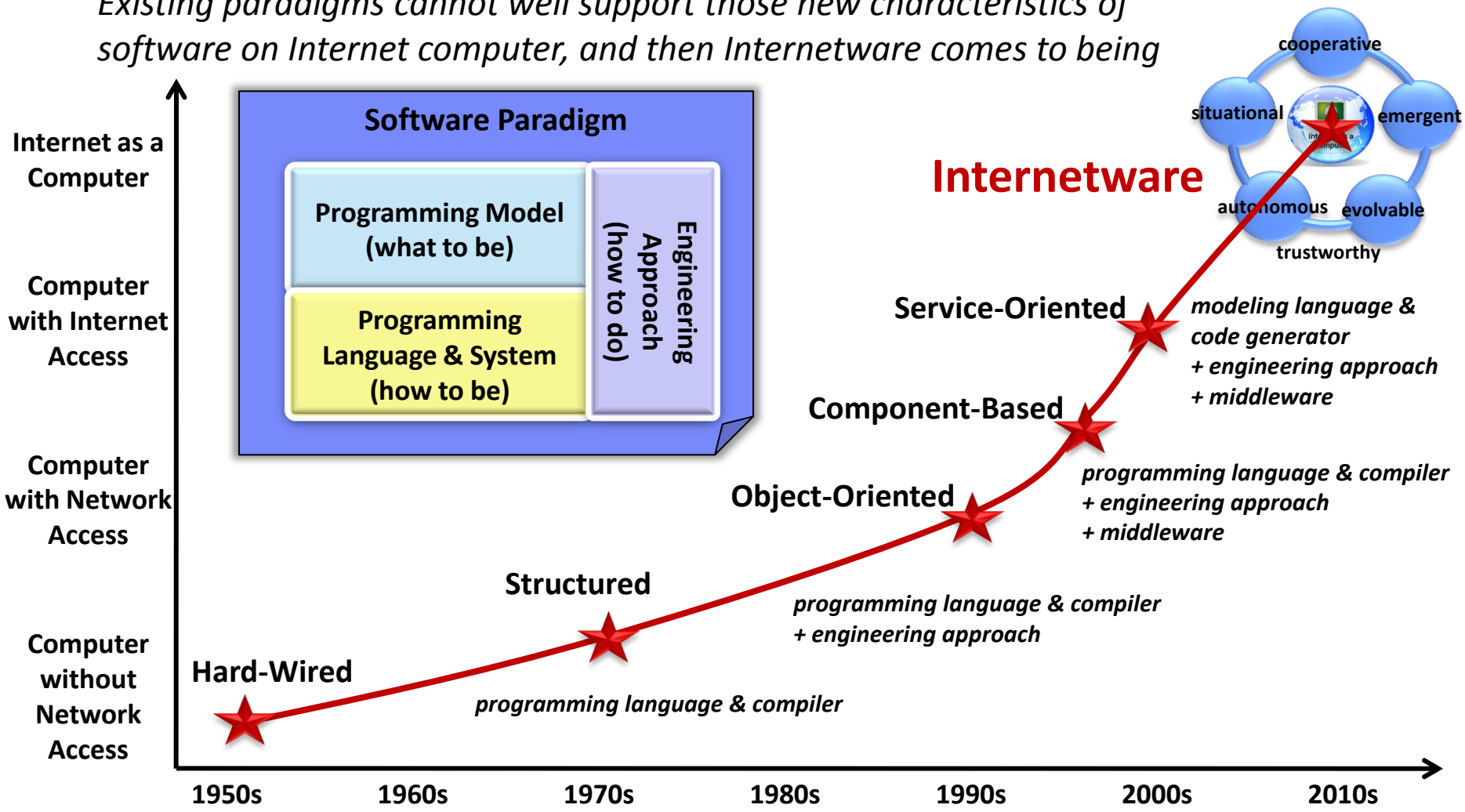
Autonomous: software is relatively independent of others; it can perform operations as it will and adapt itself when necessary

Trustworthy: software should promise some kind of tradeoff among process quality, internal system quality, external system quality and usage quality.

Internetware:

A New Software Paradigm for Internet as a Computer

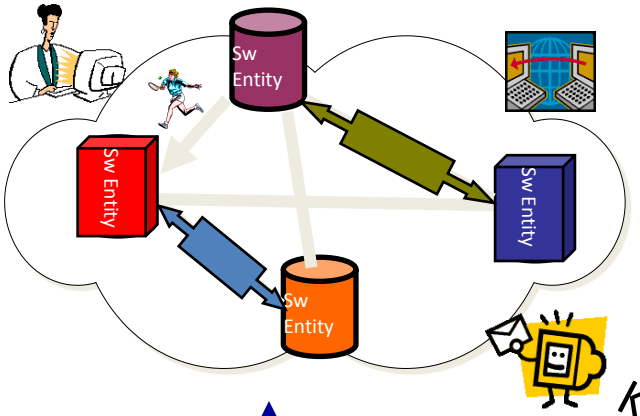
Existing paradigms cannot well support those new characteristics of software on Internet computer, and then Internetware comes to being



Software paradigm always evolves itself for evolving application domains and runtime environments

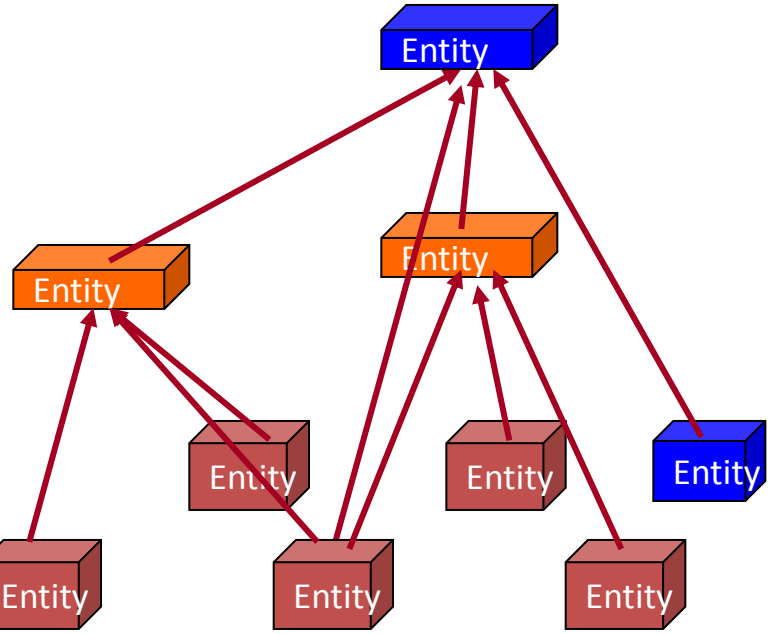
3C Modes of Internetware

On demand Collaboration



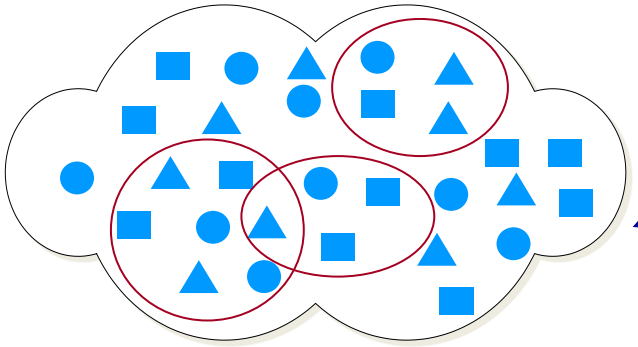
direct output

Asset and Knowledge



Behavior vs. Organization

Micro vs. Macro



Asset and Knowledge

macro output

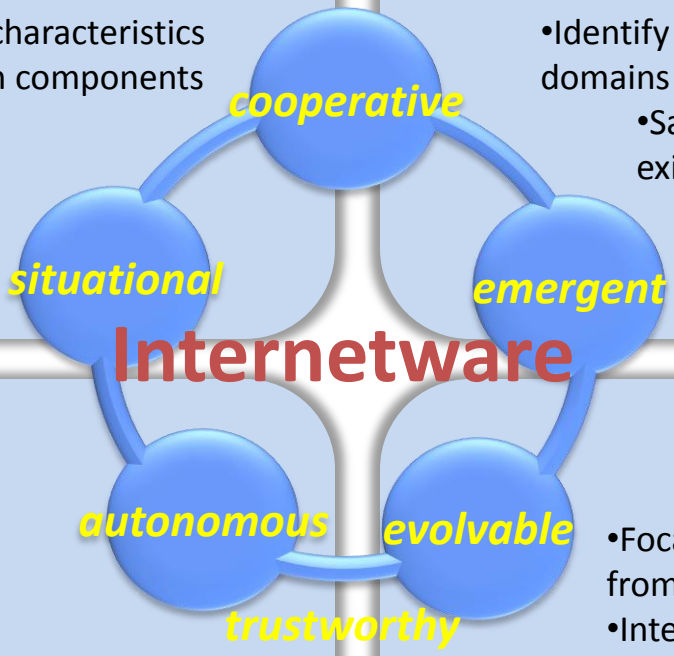
reuse-based, knowledge-driven, bottom-up, online Construction

Self-organizing Community (scope/domain)

Challenges to Internetware

Programming Paradigm(what to be)
- abstracts the elements and their relationships of a software system
- Internetware model should
- leverage legacy software and new characteristics
- Enable open collaboration between components
- Adapt itself for emergent contexts and situations

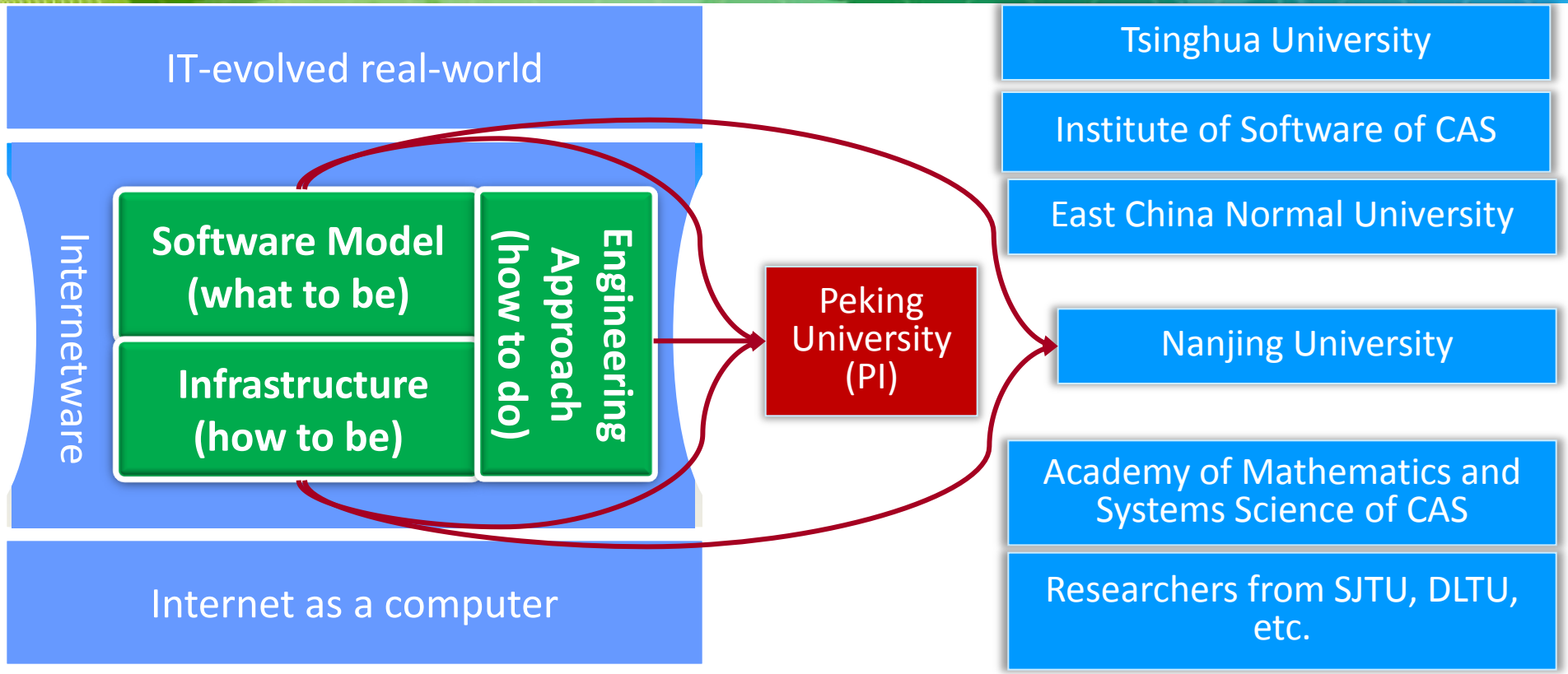
Engineering approach (how to produce)
- Systematically control the software development, deployment, maintenance and evolution
- Internetware engineering should
  - Identify the self-organized communities and domains or facilitate the self-organizations
    - Satisfy requirements via collaborating existing and/or emergent components
      - Involve all stakeholders, especially the actual end users



Programming Language & System (how to be)
- incarnates the elements and their relationships of a software model
- Internetware middleware should
- Provide a container for instantiating and operating Internetware components Provide collaboration mechanisms.
- Equip legacy software systems with Internetware characteristics
- Enable context-awareness and reflection

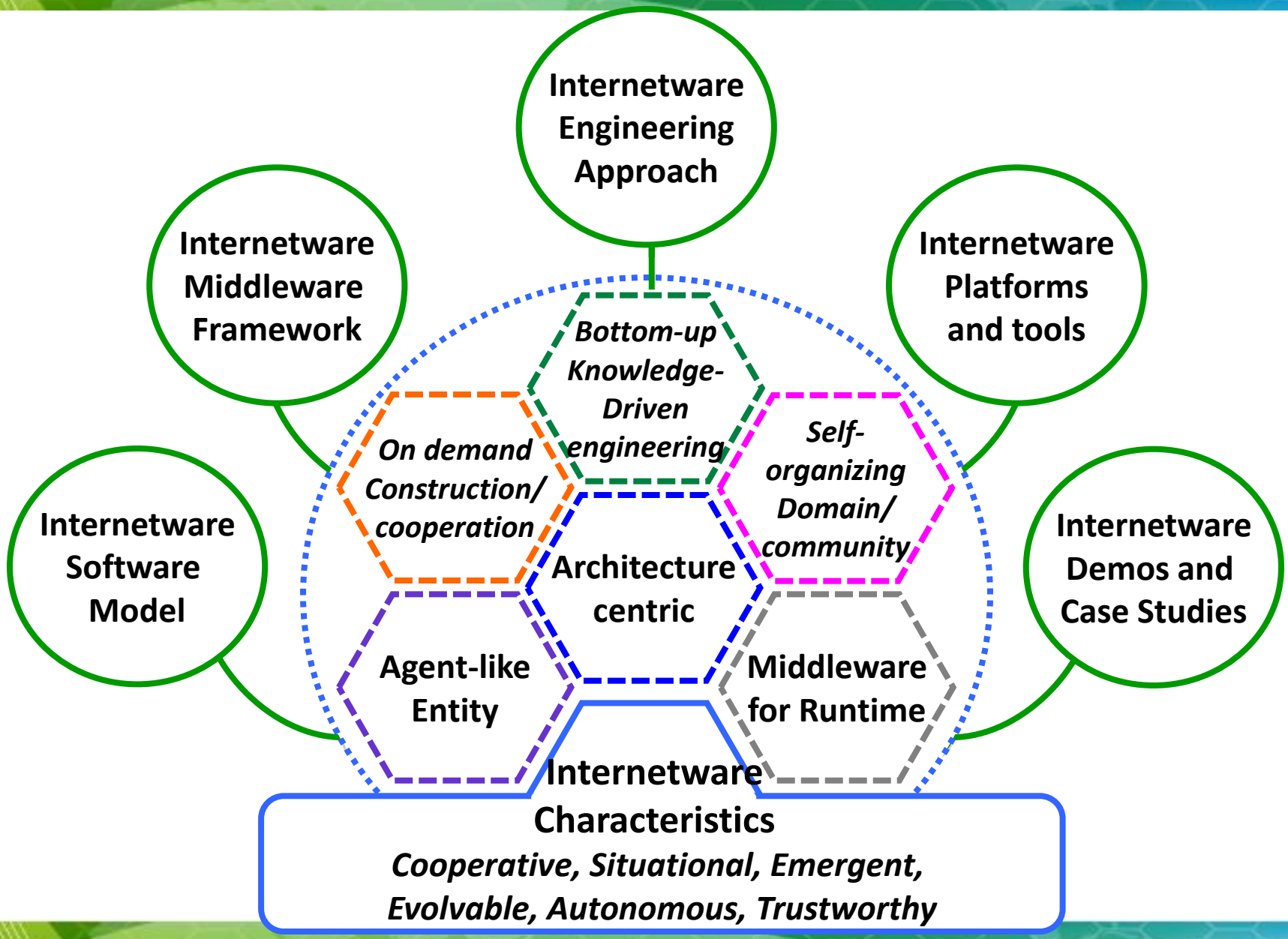
Quality assurance (how to be good enough)
- Focal points of software quality change from system-centric to usage-centric
- Internetware quality assurance should
- Define quantitative and qualitative evaluation framework for quality
- Assure the quality via engineering approach at development time as well as middleware at runtime

Internetware Research in China



- **"Theory and Methodology of Agent-based Middleware on Internet Platform"**
  - **The first national basic research program (973) project on software**
  - **From 2002~2008; > 80 faculty; 17 post-doc ; 88 Ph.D; 364 Master**
- **"High Confidence of Internetware"**
  - **IBM joined as the first foreign company in 973 program (GTO 2010)**
  - **From 2009~2013; > 100 faculty**

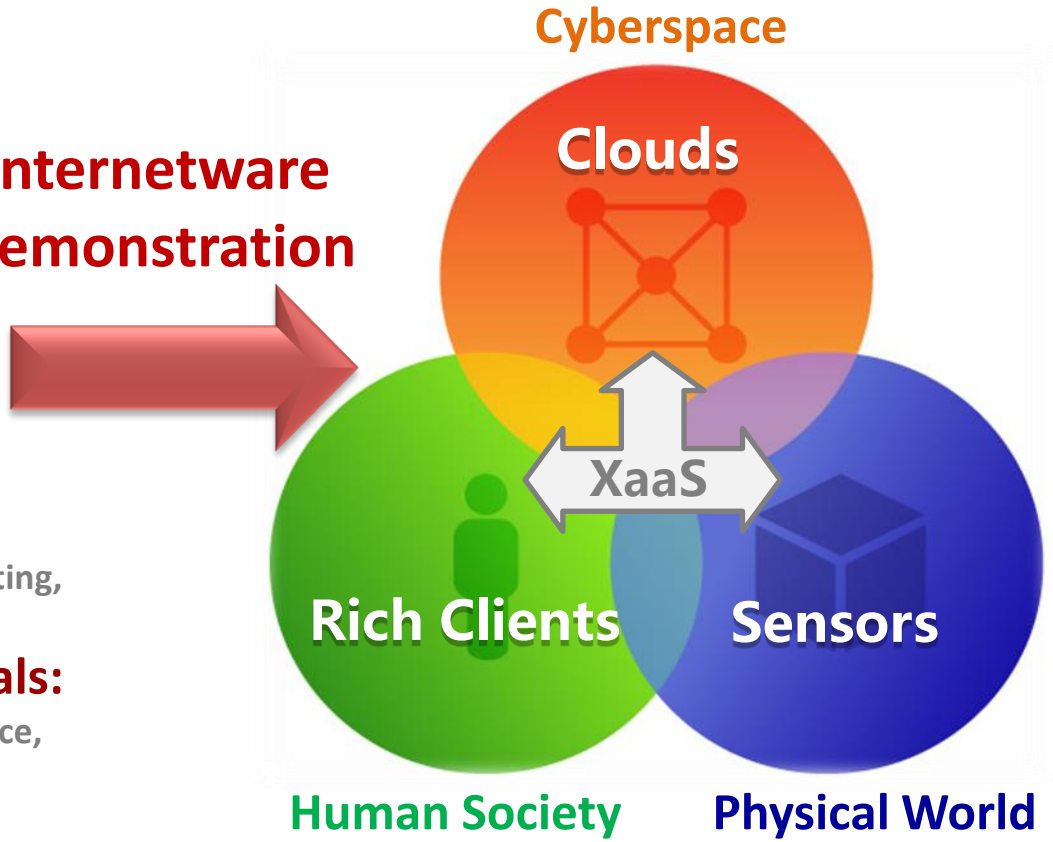
Internetware R&D Outputs



Internetware Testbed:

Synergy of computers, individuals and things on Internet

Internetware Demonstration



Synergy of Computers:

Grid Computing, Service Computing, Cloud Computing, System of Systems …

Synergy of Computers and Individuals:

Web 2.0, Virtual World, Social Network, Web Science, Social Computer …

Synergy of Computers and Things:

Pervasive/Ubiquitous Computing, Internet of Things, Cyber-Physical Systems, Smart Planet …

SM@RT (Supporting Models at Runtime)

Towards Model-based Programming Languages, Systems and Applications

Something Omitted in SaaS

Licensing challenges multi-tenancy (then new architecture)
Vendor-centric challenges delivery models (then cloud)
Customer-friendly challenges … (then …) ?

- **Software as a service (SaaS, typically pronounced [sæs]) is software that is deployed over the internet. With SaaS, a provider licenses an application to customers as a service on demand, through a subscription or a "pay-as-you-go" model. Saas is also called "software on demand." SaaS vendors develop, host, and operate software for customer use. Rather than install software on site, customers access the application over the Internet. The SaaS vendor may run all or part of the application on their hardware, or may download executable code to client machines as needed—disabling it when the customer contract expires. The software can be licensed for a single user, or group of users.**

A Client-Server View on SaaS

InternetWare

Thin → Rich Client

- Rich user experience
- Web technologies
- Browser-based
- Diverse devices

When Rich Client meets SaaS …

Rather than install software on site, customers access the application over the Internet*. The SaaS vendor may run all or part of the application on their hardware, or* ***may download executable code to client machines as needed****—disabling it when the customer contract expires.*

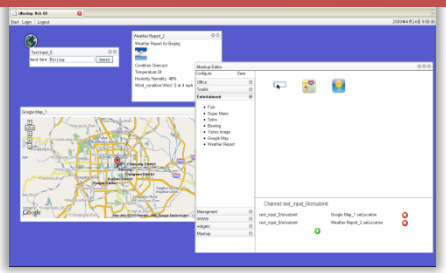
SaaS

Server → Cloud

Wide Spectrum of SaaS Rich Client

- **SaaS Rich Client is that supports service discovery, subscription, composition and even evolution in web browsers with rich user experiences**

Service Subscription	Service Discovery & Subscription	Service Composition	Service Discovery, Subscription & Composition
Web-delivered service		Web-delivered service composition	
Web APIs: Google, Amazon, Flickr, Twitter	SaaS: iGoogle, eXo, Zoho	Mashup Apps: WeatherBonk, HouseMap, Google Wave	Mashup Environment & Apps: iMashup, Palm WebOS



How to synthesize services with rich user experiences

How to handle diverse client devices and platforms

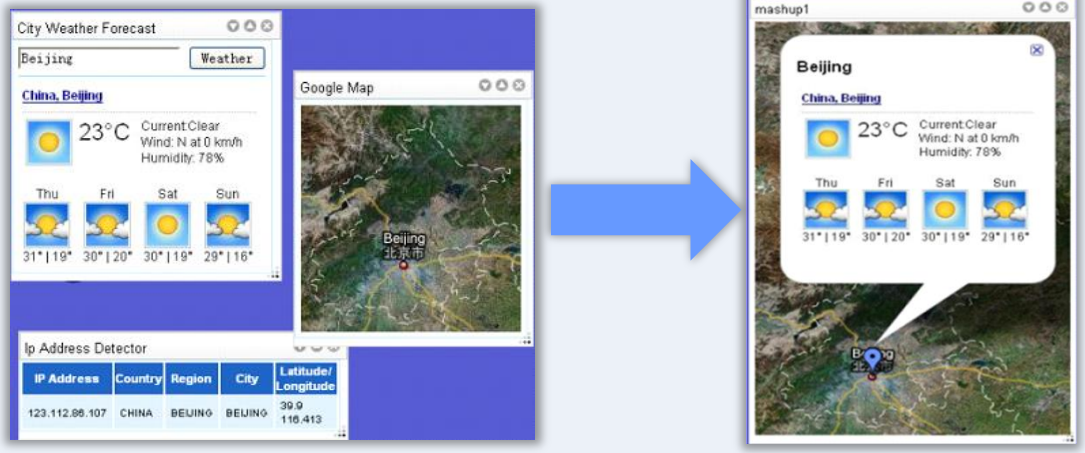
Simple, Easy-to-use ⟷ **Powerful, Professional**

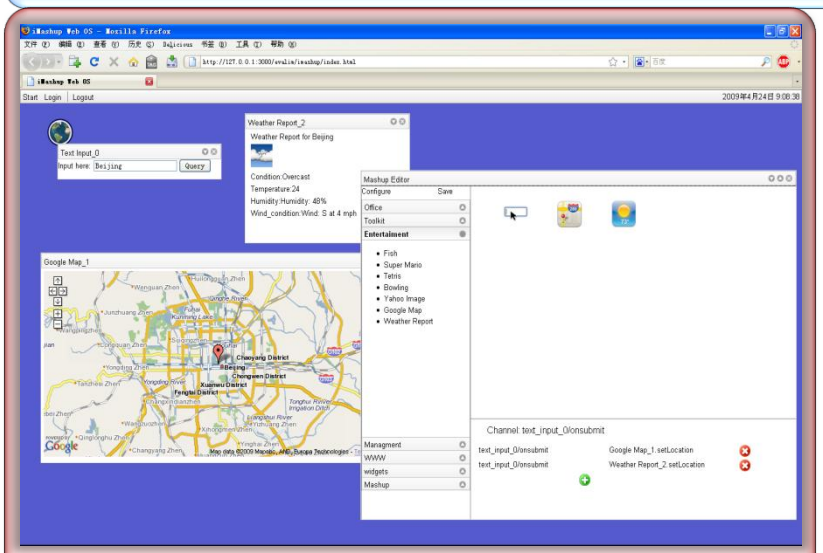
Synthesize Services with Rich User Experiences

InternetWare

Rich Client Component

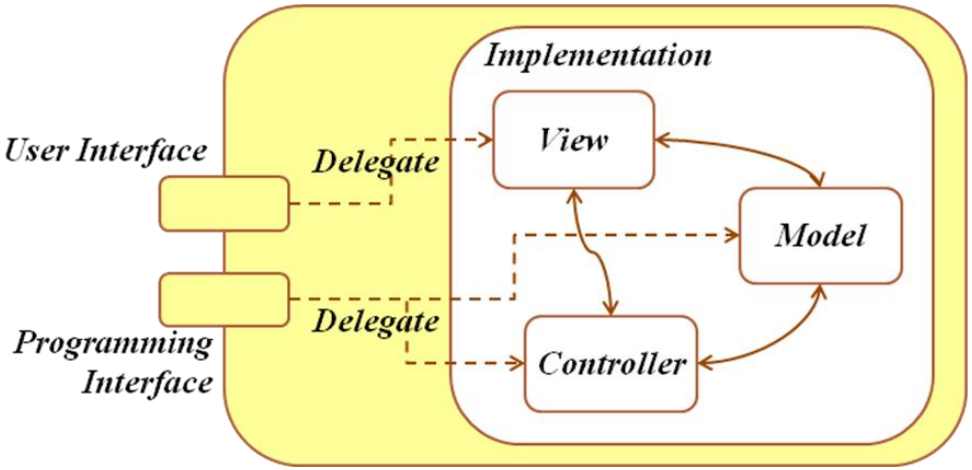
Rich Client Component Composition



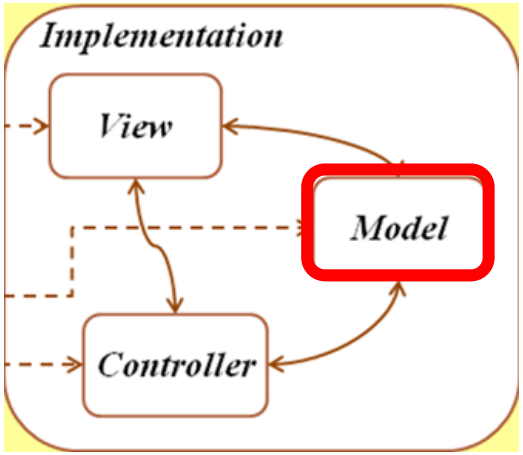


On-the-fly Qualification of Component & Composition

new client-side programming model is needed



Handle Diverse Client Devices and Platforms

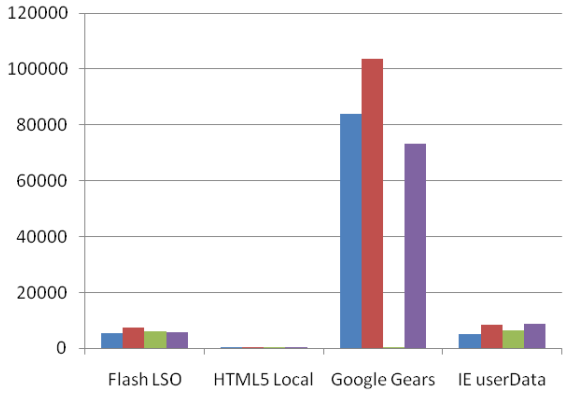


Model Data from Local Storage

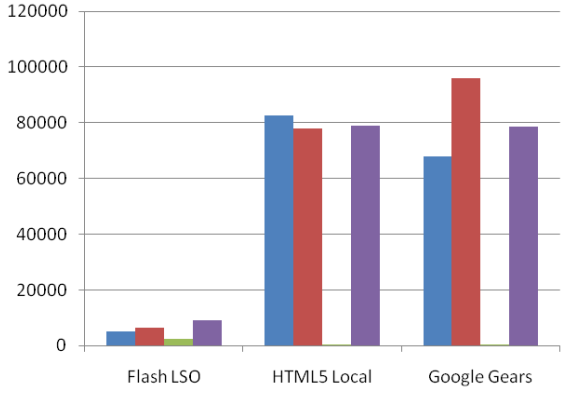
Adapt to heterogeneous local storage supports

	IE	Firefox	Chrome	Safari	IE (Mobile)	Safari (iPhone)
IE userData	5.5+	N/A	N/A	N/A	N/A	N/A
Flash LSO	Plug-in	Plug-in	Plug-in	Plug-in	N/A	N/A
HTML5 Local	8.0+	3.5+	3.0+	3.1+	N/A	3.1+ (OS 2+)
HTML5 DB	N/A	N/A	3.0+	3.1+	N/A	3.1+ (OS 2+)
Google Gears	6.0+,Plug-in	1.5+,Plug-in	1.0+	N/A	N/A	N/A

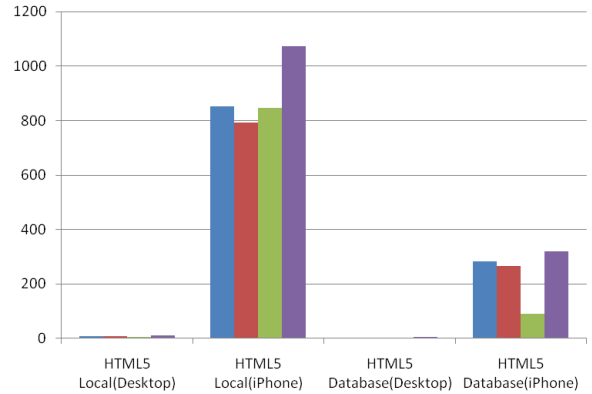
Select the best-of-the-breed local storage



IE 8



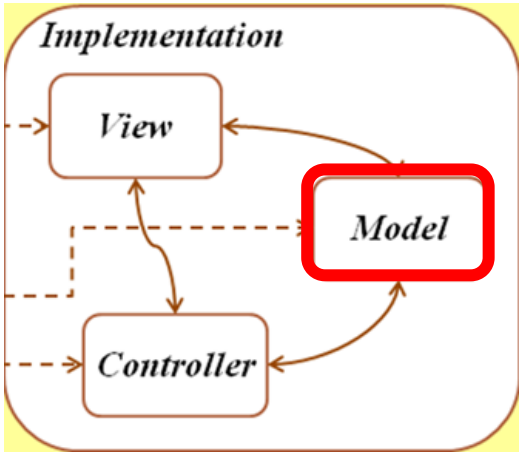
Firefox 3.6



HTML5 Performance in Safari Desktop & Mobile

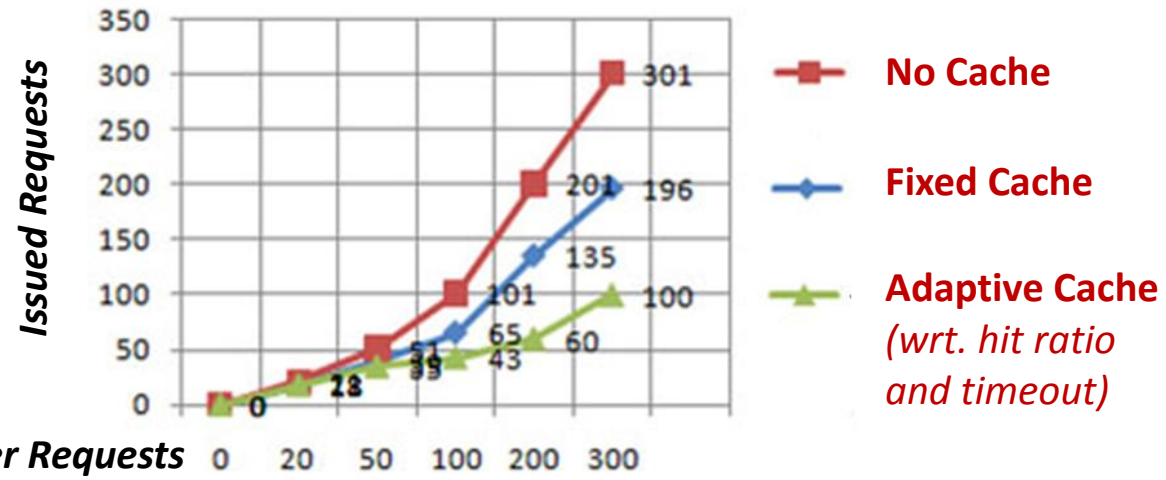
Create, Read, Update, Delete Performances

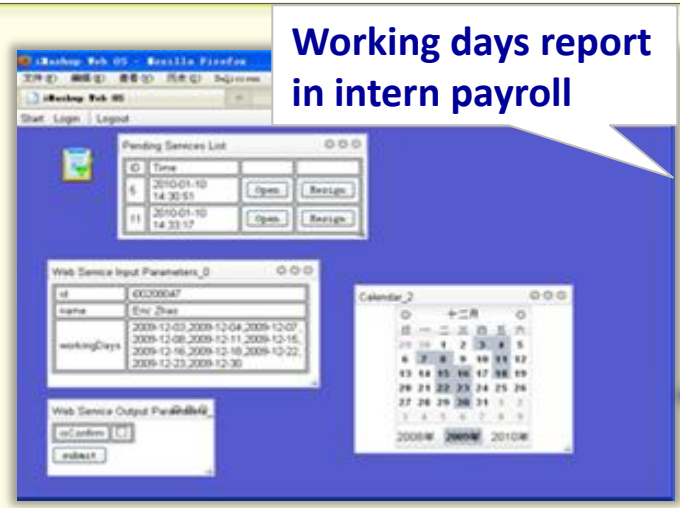
Handle Diverse Client Devices and Platforms



Model Data from Remote Service

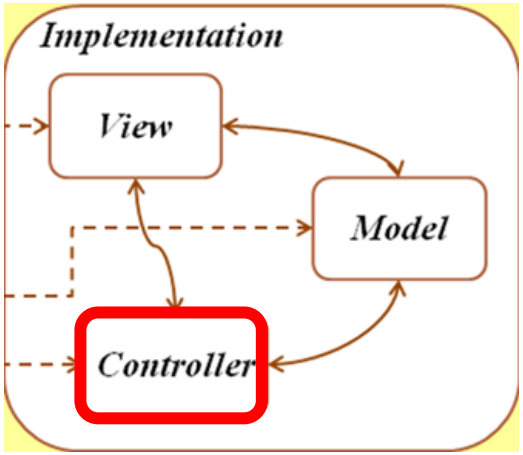
Data Cache of Remote Service (online/offline)





Transmit data between Rich Client and (remote) Business Process

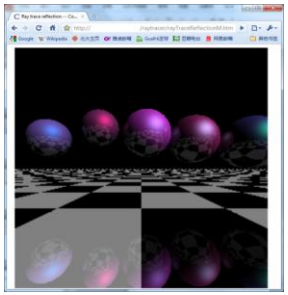
Handle Diverse Client Devices and Platforms

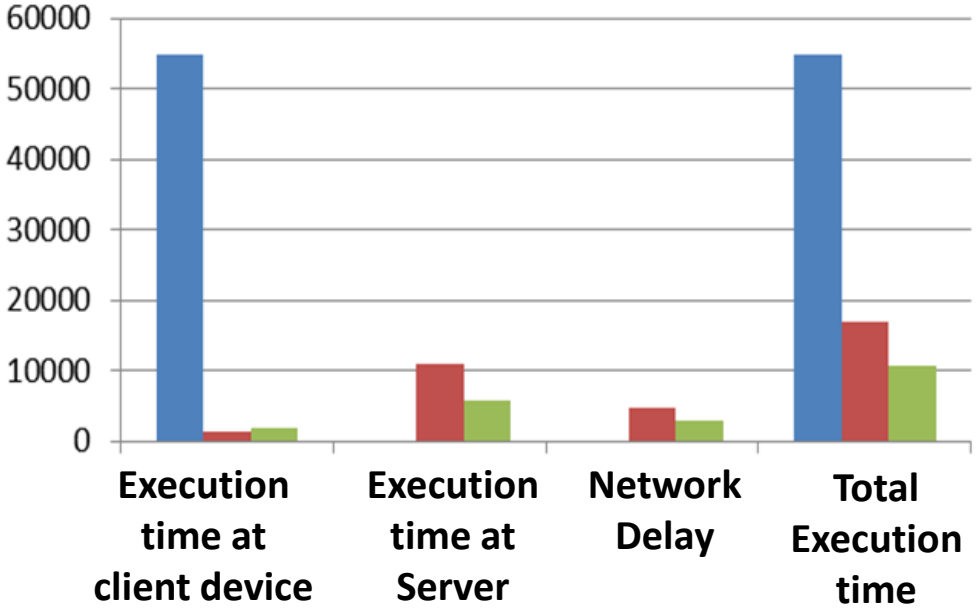


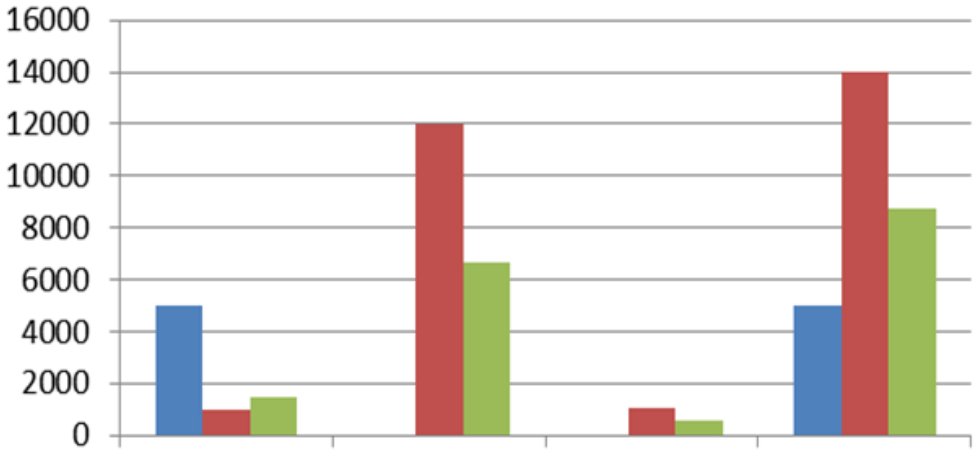
Controller adapt to Client Devices

- All at the client device
- Code back to server from client
- Code at both server and client

**Language dynamism (Javascript) is challenging*





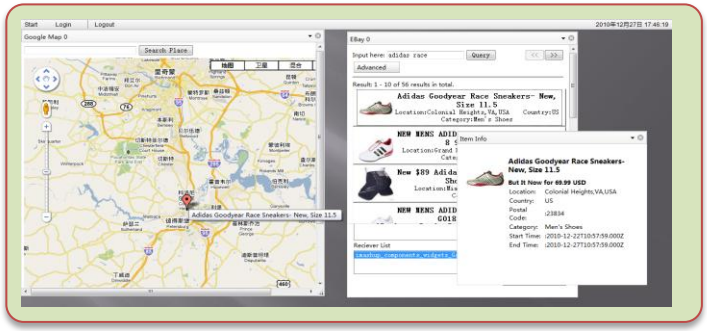


**Chrome V8 at server can drastically speed up IE, Firefox and other browsers at client (even on PC)*

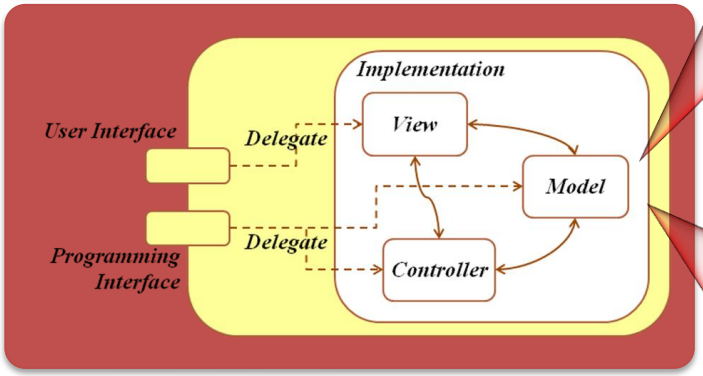
SM@RT Clients:

An MVC-Oriented Component-Based Programming Framework (in Javascript)

SM@RT Mashup Tool & Apps



SM@RT Client Framework



Web Stack

OS

Device

On-the-fly Composition:

@MVC Component: create instances at the time when a component is selected

@MVC Composition: propagate messages at the time when a connection is specified

@Quality Assurance: design at runtime; model checking at runtime

Adaptation for Heterogeneity:

@Model:

- **local/remote/mobile storage;**
- **online/offline (consistency)**

@Controller:

- **phone/tablet/PC computing power**

@View: different screen size

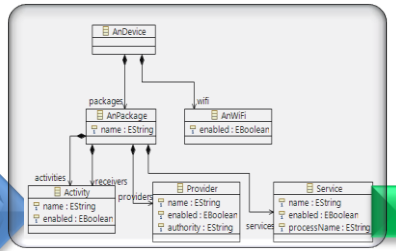
SM@RT Clouds:

A Model Driven Framework for Autonomic Management

- *Synchronizer*: causally connecting MOF-compliant models to runtime systems so that changes on one side will cause corresponding changes on the other side just-in-time
- *Synthesizer*: integrating, orchestrating and applying multiple user-defined model analysis methods on runtime models

Trivial, scattered and tangled management-related data is accessible via system APIs and files (scripts & configurations)

gather data just-in-time



organize to MOF-compliant models

Feed is-be models to analysis & decision tools

Petri-Net SPIN CSP Z LTS

Effectuate & integrate analysis results

MOF and QVT compliant, Non-intrusive, Automated

Change data according to to-be models

change

query, transform & synthesize

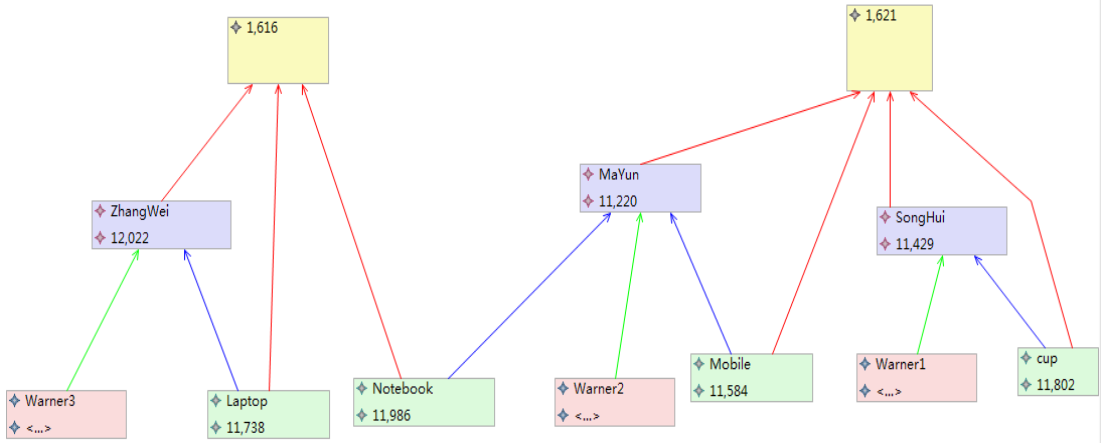
Visualize

SM@RT Sensors:

An Object/Sensor Mapping Framework in Java *(just like ORM)*

```
public void OnRecvTag(object sender, RecvTagEventArgs e)
{
    if (controlPane.IsStopRecvTagID) return;

    lock (lstRecvData)
    {
        lstRecvData.Add(e);
    }

    ShowTagIDHandle d = new ShowTagIDHandle(ShowTagID);
    this.BeginInvoke(d);
}
```

55946	10131	-57.5	-20	2	false	2010/11/30 22:08:31
55419	10131	-59.5	-20	2	false	2010/11/30 22:08:31
55070	10131	-58	-20	2	false	2010/11/30 22:08:32

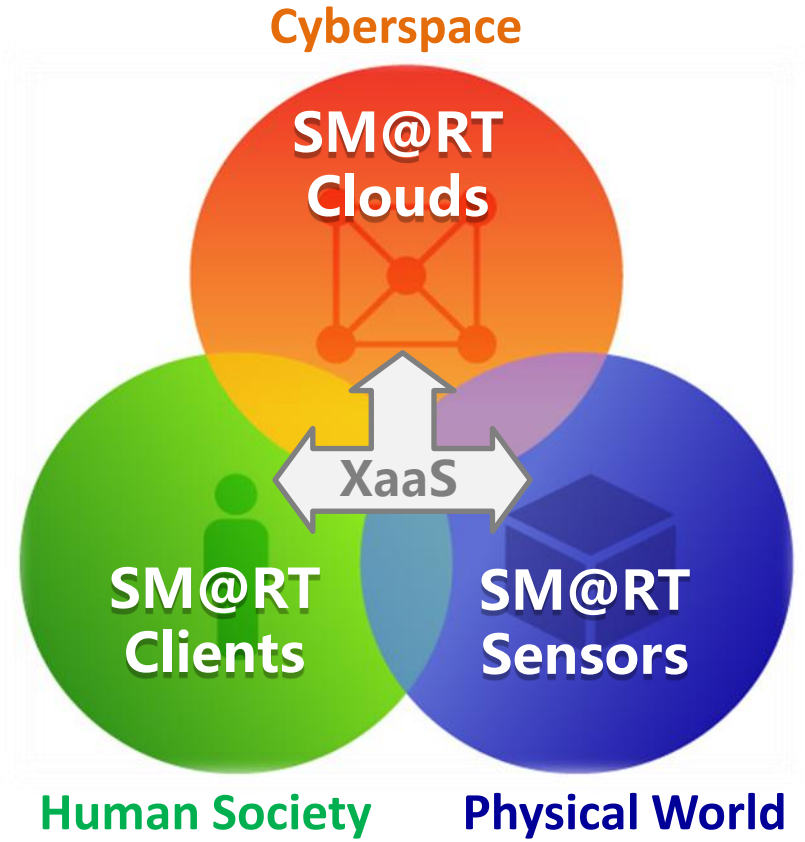


Transform (in a model-driven way) sensor-specific data and functions to application-specific objects

```
public class HouseImpl extends EObjectImpl implements House {
    /**
     * The default value of the '{@link #getId() <em>Id</em>}' attribute.
     */
    protected static final int ID_EDEFAULT = 0;

    /**
     * The cached value of the '{@link #getId() <em>Id</em>}' attribute.
     */
    protected int id = ID_EDEFAULT;

    /**
     * The default value of the '{@link #getTemperature() <em>Temperature</em>}' attribute.
     */
    protected static final String TEMPERATURE_EDEFAULT = null;

    /**
     * The cached value of the '{@link #getTemperature() <em>Temperature</em>}' attribute.
     */
    protected String temperature = TEMPERATURE_EDEFAULT;
```

Internetware

A Software Paradigm for Internet as a Computer



Internetware Testbed: Demonstrate the Synergy of computers, individuals and things on Internet