Ubiquitous Web Applications

Dave Raggett, W3C

Contact: dsr@w3.org

Why Standards?

- Standards are expensive and time consuming to create, why bother?
- Large and small companies may feel they can just develop their own solutions, much easier!
- But standards encourage a bigger market with many more players and more innovation
- That means that everyone wins
- Users are no longer in thrall to single vendors

W3C

- International consortium founded in 1994 with a mission to lead the Web to its full potential
- Directed by Tim Berners-Lee, inventor of the Web
  - Initial project proposal in 1989 to CERN
- Over 400 members from all across the World
- Hosted by Keio University in Japan, ERCIM in Europe and MIT in North America
- Over 60 staff members
- 17 regional partners to promote W3C work

W3C

- W3C has produced over one hundred Recommendations covering HTML, XML, CSS, Web Services, Semantic Web and many more
- Open process and patent policy designed to enable royalty fee implementations of W3C specifications
- 47 Working Groups, 12 Interest Groups, 4 Coordination Groups, 4 Incubator Groups, Technical Architecture Group, Advisory Board, and the Advisory Committee with one representative from each W3C Member

Brief history of my involvement

- Studied physics/astrophysics at Oxford
- HP Labs, working on knowledge-based systems
- Hypertext-based expert system for generating quotes for HP computer systems
- Started working with TBL on WWW in 1992
- HTML+, HTML 3.0, HTML3.2, HTML4, XHTML
- HTTP, Math, Forms, Voice, Multimodal and now the Ubiquitous Web

Ubiquitous Networked Devices

- In 1965 Gordon Moore (Intel) predicts doubling of components on silicon chips every 2 years
- Today a single chip may have hundreds of millions of transistors and run at GHz rates
- Silicon radios – combining computers with RF signal processing
- Ubiquitous cheap digital device controllers
- RFID chips that fit within the groove of a finger print
- Very low cost to add networking to all devices

Evolving Networking Technologies

- Ethernet over twisted pair or coax
- DSL over copper phone lines
- Ethernet over building power wiring
- WiFi
- Bluetooth
- GSM and cellular packet radio
- WiMax
- *An ever changing choice of technologies*

Examples of Devices

- Security sensors for movement, pressure, windows/doors
- Door locks and security cameras
- Smoke, Carbon Monoxide and pollution detectors
- Lighting, heating and other environmental controls
- Household appliances (e.g. washing machine, freezer)
- Hand held remote controllers
- Flat screen display/television sets
- Media servers and Home gateways
- Phones, Printers, Scanners, Cameras, Projectors, ...
- Devices in cars, trains, ships, and planes ...

What's the Value?

- Improved physical security and peace of mind
- Reduced costs of heating/cooling/lighting homes and offices
- Preventative maintenance in advance of appliances breaking down
- Better choices for home entertainment systems
- Access to information services any time, any where and on any device you choose
- Fulfilling the potential for applications that combine local and remote services

What do people think the Web is?

- Most people think of the Web as something you access from a browser on a PC
  - Big colourful high resolution screen
  - High speed connection
  - Mouse pointer and full sized keyboard
- Limited awareness of accessibility problems
- Virtually no awareness of relevance to other kinds of devices and modes of interaction
- Yet, voice interaction is growing rapidly as Web technologies are applied to call centres

Aren't current standards sufficient?

- Lots of people are building web applications using HTML with lots of client and server-side scripting
- This is expensive and very specific to desktop browsers with poor user experience on mobile devices
- Ajax is cool, but too low a level of abstraction
- The same is true for Web Services
- Very limited access to local device capabilities
- Inadequate for harnessing ubiquitous devices

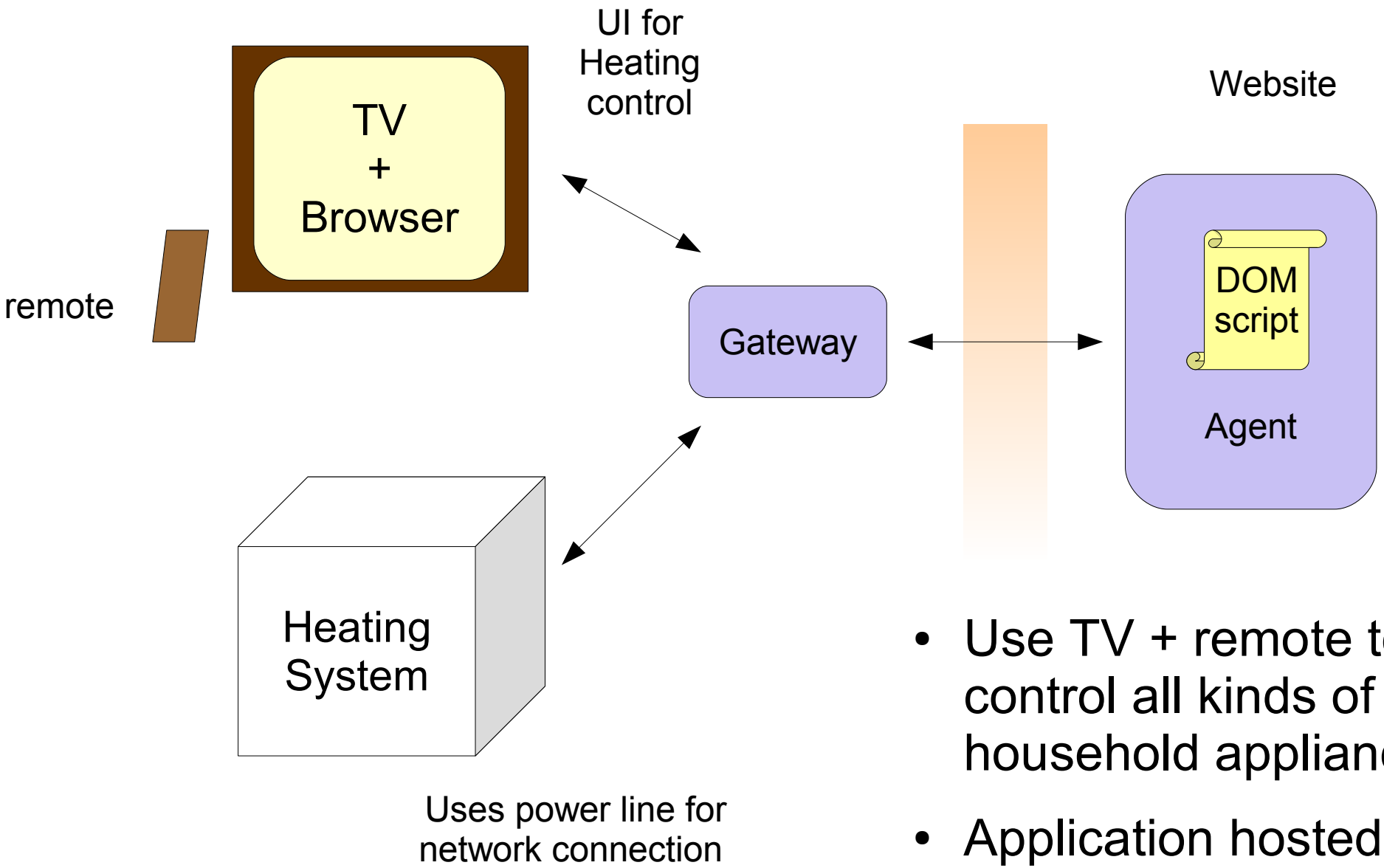
Security and Privacy Concerns

- The Web is a mess when it comes to security
- Different user name/password for each website encourages people to use weak passwords
- Wide open to phishing attacks
- Criminal gangs harnessing compromised PCs to send out spam and to launch attacks
- Privacy abuses are commonplace
- Browser sandbox model and same-site policy are too weak and work-arounds introduce major security/privacy holes

Trust Management Solutions

- Users tend to click through security related dialogues that “get in the way” of the task
- Users are often not really informed about the trustworthiness of a website/application
- We need to find solutions that offer greater security with improved usability
- Improved security through SIM cards and biometric techniques
- Trust delegation solutions involving a trusted third party

Home network example



- Use TV + remote to control all kinds of household appliance
- Application hosted by website

Realizing the Potential

- Initially, just proprietary solutions
  - end user purchases complete solution
  - single vendor and single product generation
- Followed by narrowly focused industry standards
  - e.g. Pictbridge as solution for printing direct from camera when printer and camera from different vendors
- Broader standards follow later, enabling new applications
  - Traditional programming languages like C++ and Java offer low level control but are costly to develop with
  - Web technologies will make applications easier and cheaper to develop, enabling a much bigger ecosystem

What's needed to achieve this?

- Standard-based architecture that decouples application authoring from the details of networking technologies and device platforms
- Standards for groups of devices with similar functions so that applications are not tied to specific devices
  - Bringing together interested parties to work on ontologies of device capabilities and exposure as APIs for markup and scripts to access these capabilities
  - Careful consideration for versioning to ensure that new devices will work with existing applications, and that new applications will work with older devices

How is W3C addressing this?

- New Ubiquitous Web Applications Working Group
  - Launched 30 March 2007
  - Successor to former Device Independence WG
  - Broadened focus on Ubiquitous Web Applications
- Support for regional subgroups
  - can hold meetings in local language, e.g. Japanese
  - meeting summaries and technical specs in English
- Balance between openness and confidentiality
  - publish approved meeting summaries and approved editorial drafts of technical documents

UWA Approach

- Define user interface, data models and behaviour as combination of markup and event-driven scripting
  - XML + Events + RDF + Object Model
- Device coordination framework
  - descriptions, binding and use of capabilities
  - support for rich meta-data and trust delegation
- Logical support for passing events between devices over different networking technologies
  - coupling devices and support for remote user interfaces
- Distinction between authoring and execution
  - policy-based content adaptation to match the delivery context (user preferences, device capabilities, etc.)

Device Behaviour

How to "program" device behaviour?

- Simple devices with fixed behaviour
- XML + scripted event handlers
  - e.g. XHTML/SVG + ECMAScript
- Pure XML with language defined event handlers
  - e.g. SCXML (StateChartXML)
    - event driven state machines as in UML
- Pure script with event handlers
  - Device has script engine + library of objects

Device Coordination Framework

Finding and binding to services
in the context of an application session

Examples of Services

- Device capabilities, e.g.
  - audio capture and playback
  - embedded camera
  - ability to initiate a phone call
  - persistent storage
  - calendar, address book, personal preferences, ...
- Speech synthesis and recognition
  - using embedded or remote speech engine
- Geographic location

“service” is used loosely for anything that Web applications might want to make use of

Binding to a Service

- Binding as a scripting interface
  - Input a service name or description
  - Output an object that proxies for the service
- May be restricted and based upon proving membership of appropriate access control list
  - Issues of trust, identity, privacy and security
  - Usability issues, e.g. asking user for decision
    - Is it okay to send location to web app?
    - Is it okay to grant access to camera?
    - What information to provide as context?
    - What if user isn't present?

Service Discovery

- Name service or describe its characteristics
  - URI for service or service description
  - Description as content for XML element that will act as DOM proxy for the service
- Discovery mechanism may be implicit
  - Provided by run-time environment, e.g. UPnP
- Discovery mechanism may be explicit
  - Provided by a named Web server
  - Based upon external description of service

Binding as Markup

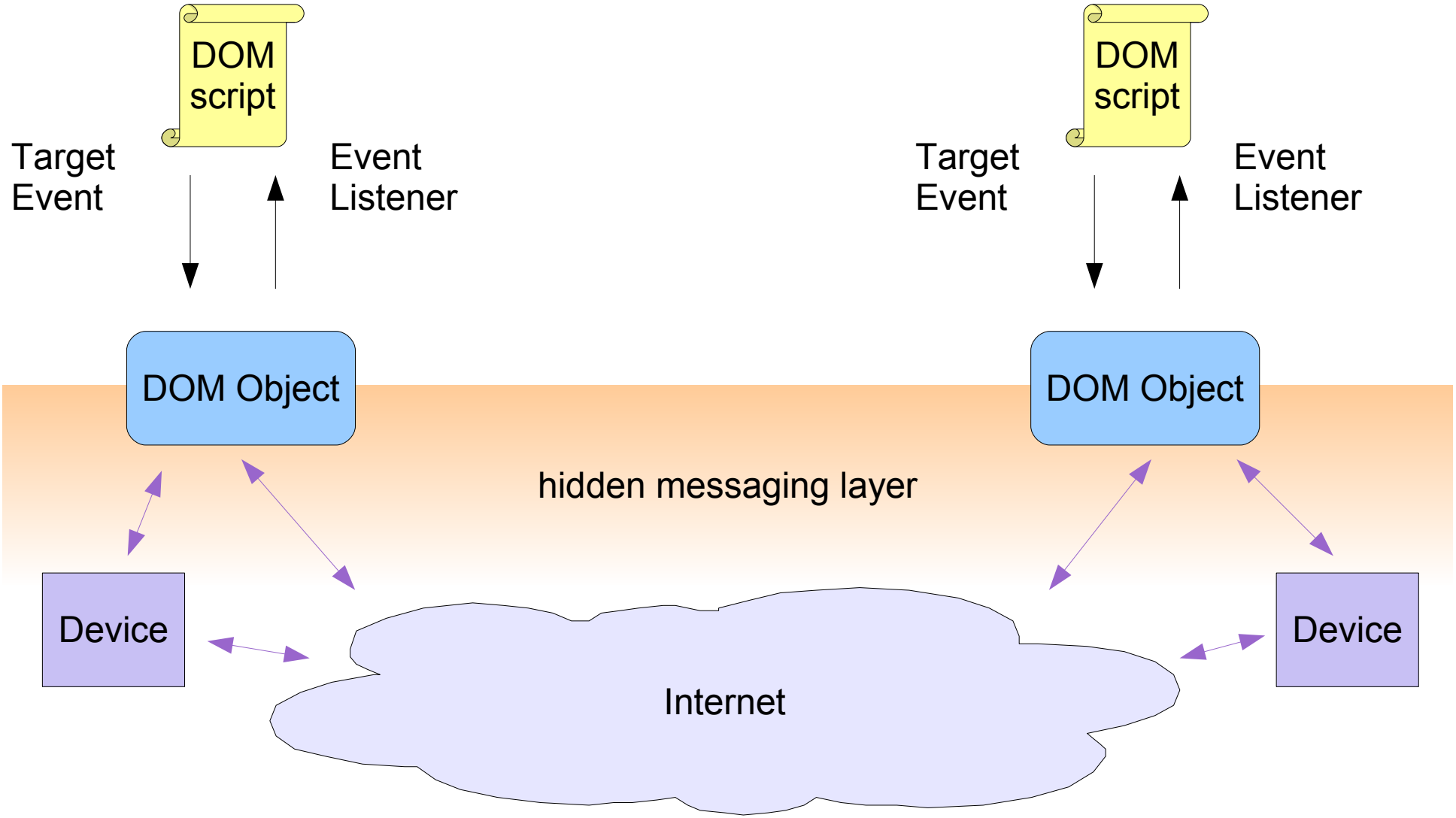
- Markup element as proxy for service/capability
  - Attribute that names service/capability as a URI
  - Or URI for reference to external description
  - Or content markup as description
  - When binding is complete, raises binding event
  - Or error event if binding fails if access is denied
    - fallback markup for an alternative
  - Another event when resource is unbound
- Target events at element to control resource
- Set event handlers to respond to changes

Delivery Context Client Interfaces

- Enable applications to dynamically respond to changes in user preferences, device capabilities and environmental conditions
- Exposed as tree of XML DOM Nodes
  - For example, display characteristics, playback volume level, memory size, geographical location, battery level, network availability, etc.
  - Nodes may support additional interfaces for accessing services, e.g. dimming display, or muting microphone
  - Nodes act as proxies for accessing capabilities

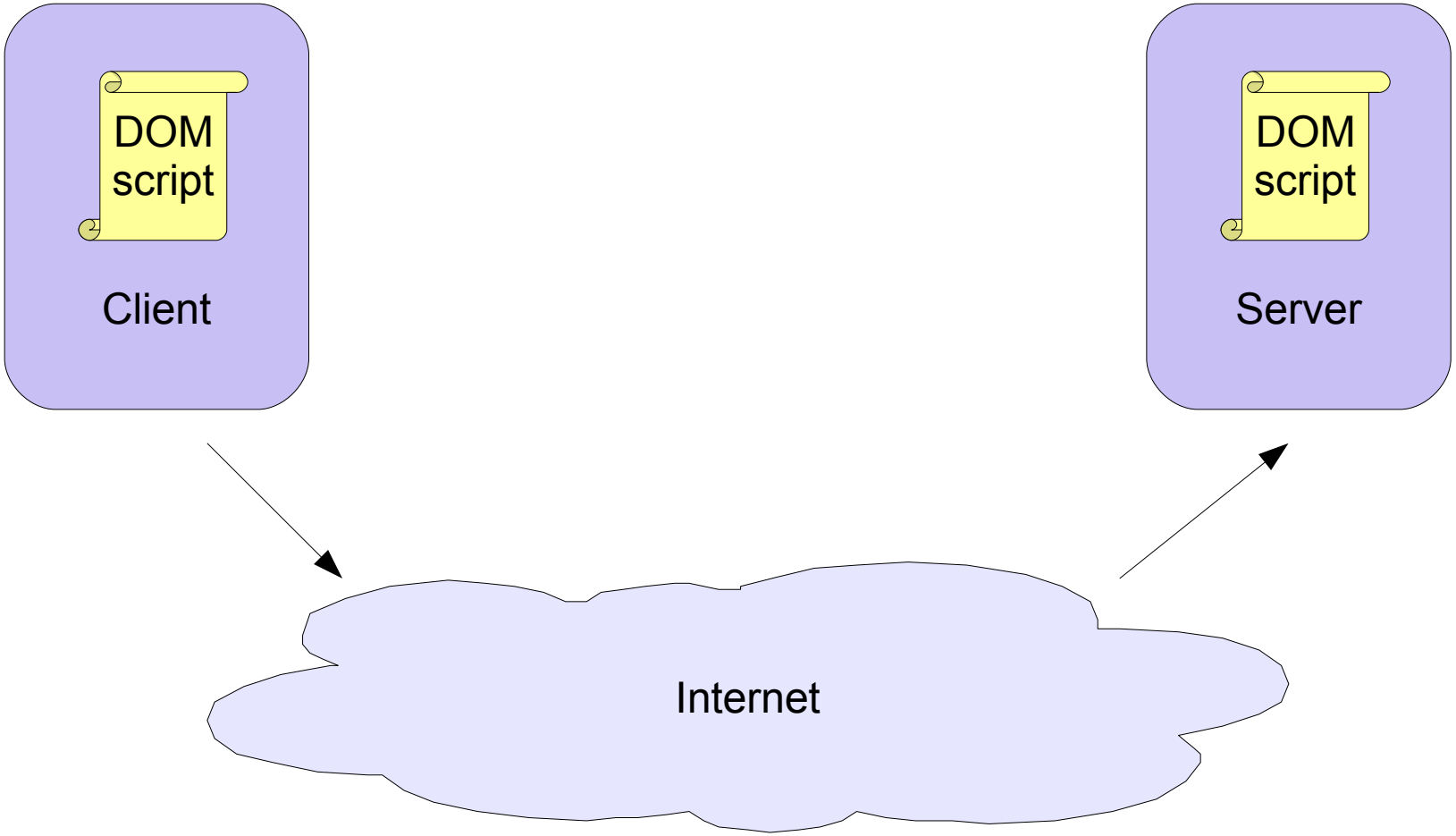
DOM = Document Object Model

Proxies for accessing services



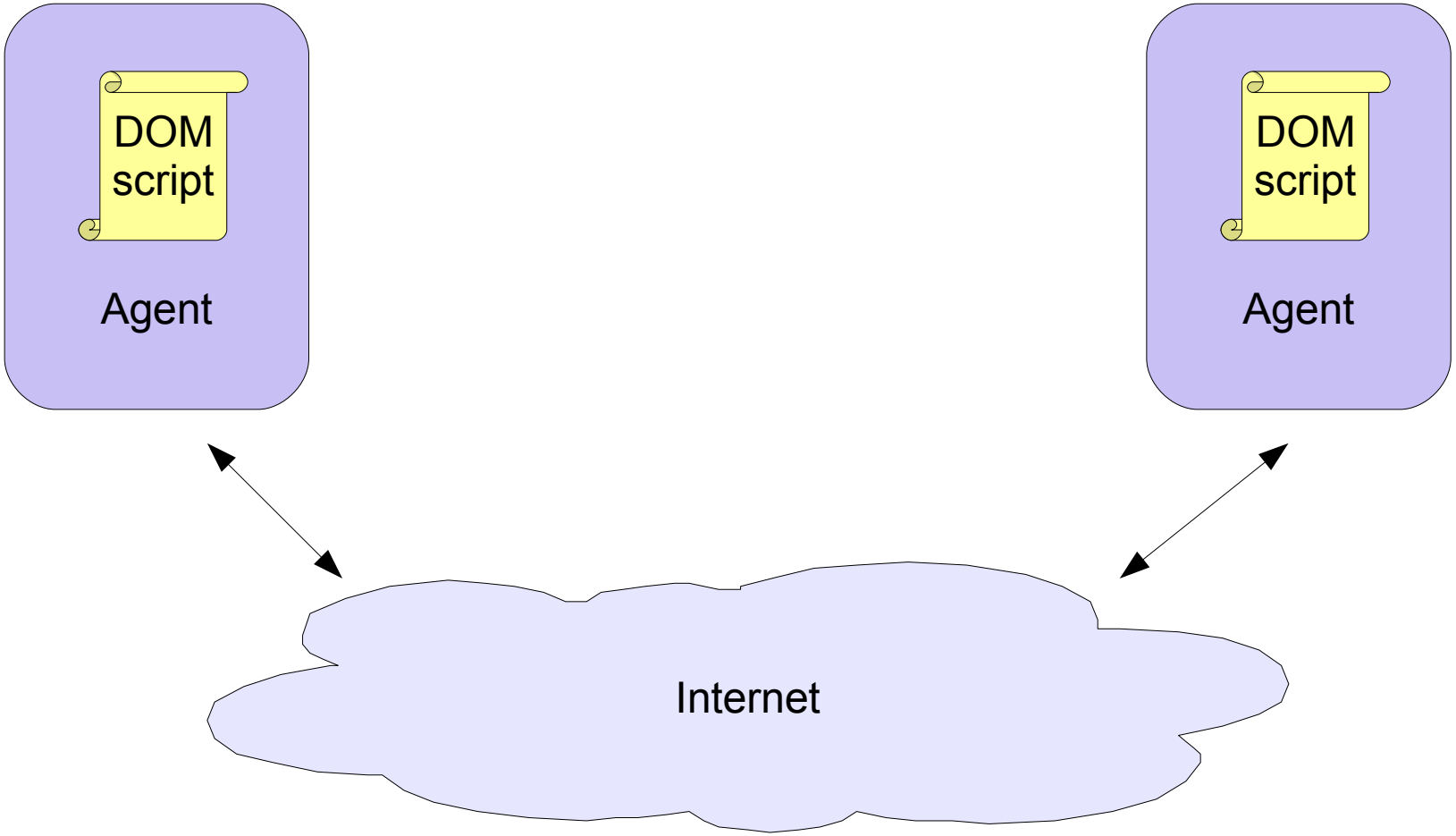
DOM – XML Document Object Model

Client or Server?



Client or Server?

Agent combines client and server

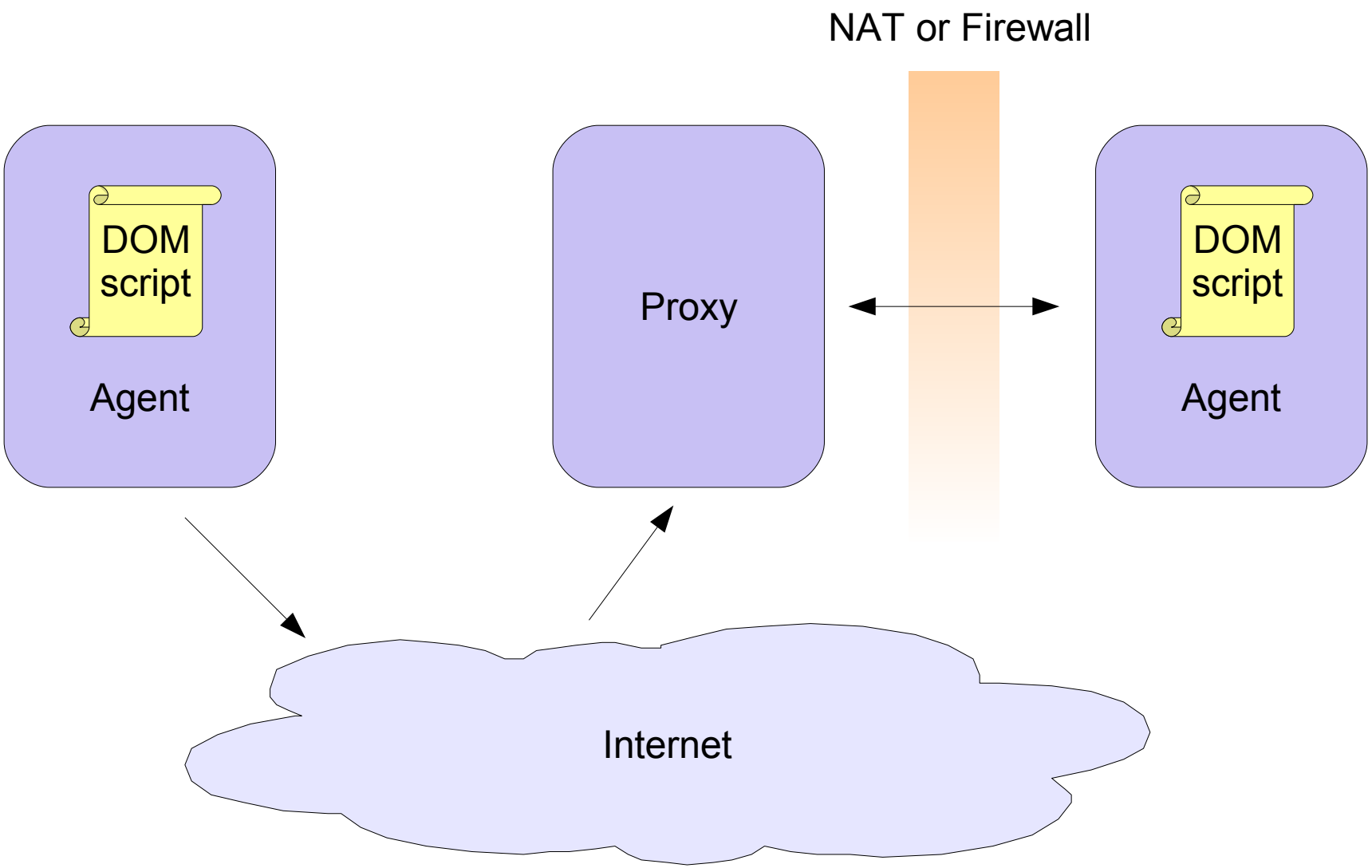


Event Transport

How to deliver events to devices?

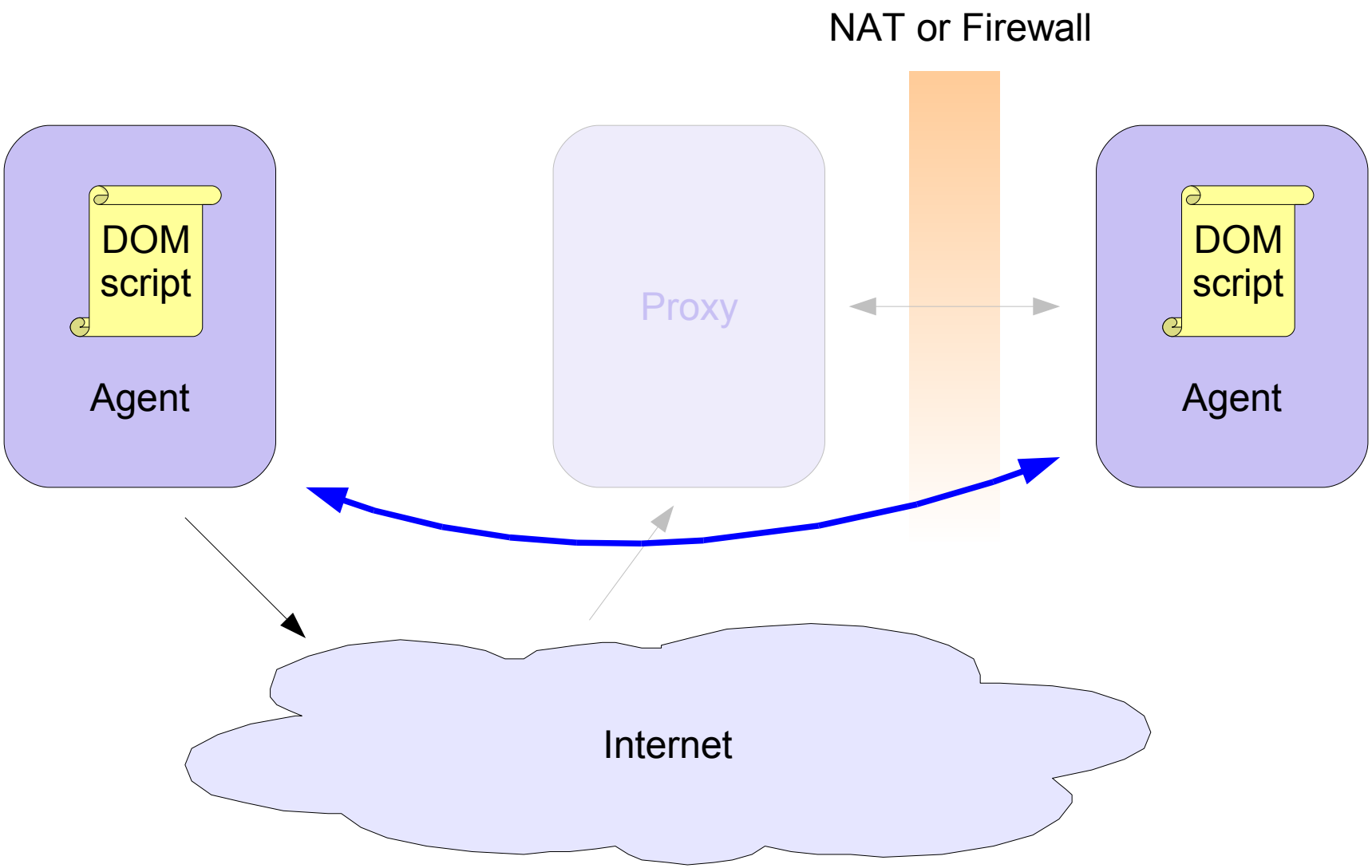
- HTTP
  - a) Add HTTP server to each device
    - But problems with firewalls/NAT
  - b) Emulate via polling/long lived connection
    - Hacks with Ajax
- Overloading SMS on GSM networks
- SIP and IMS
  - Each device acts as client and server
    - IETF/3GPP standards
    - XML representation of event as SIP message payload
    - But see also IMPP as used for Jabber and Google Talk

Tunnelling through NAT



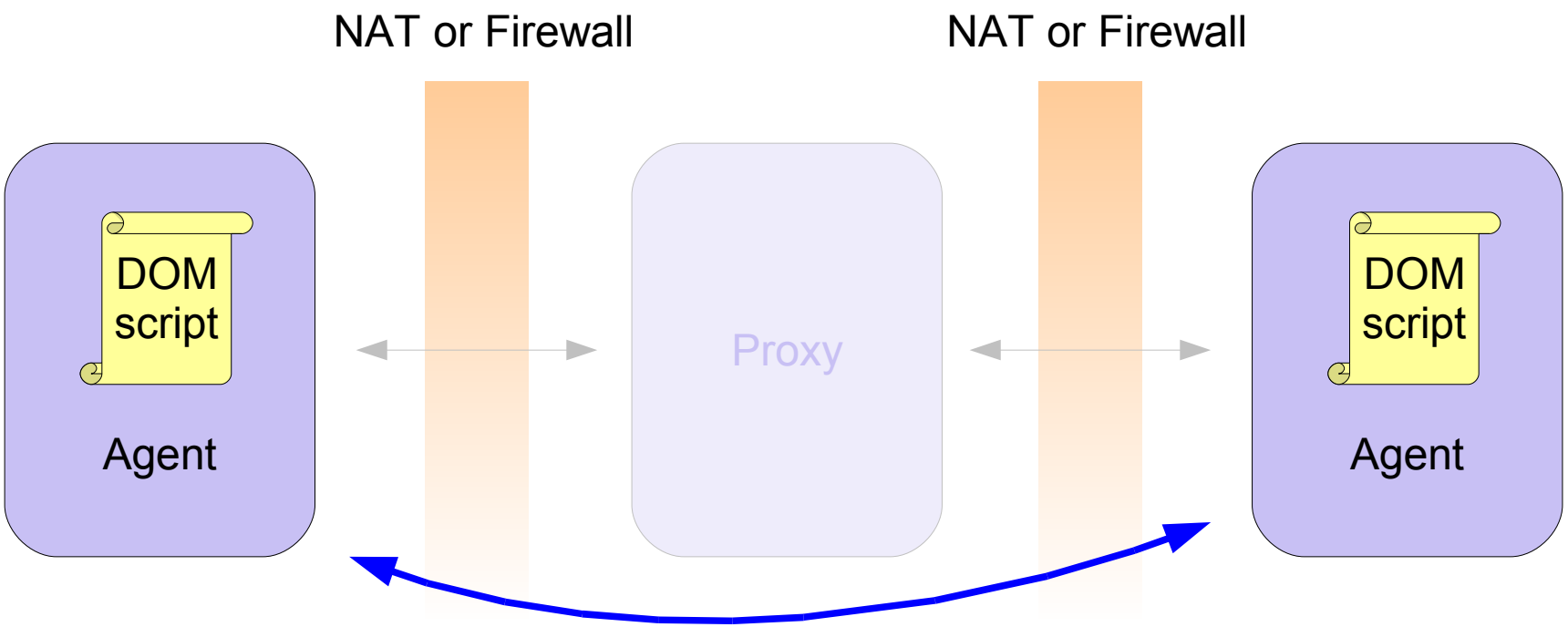
Tunnelling through NAT

Proxy may arrange for direct link through NAT



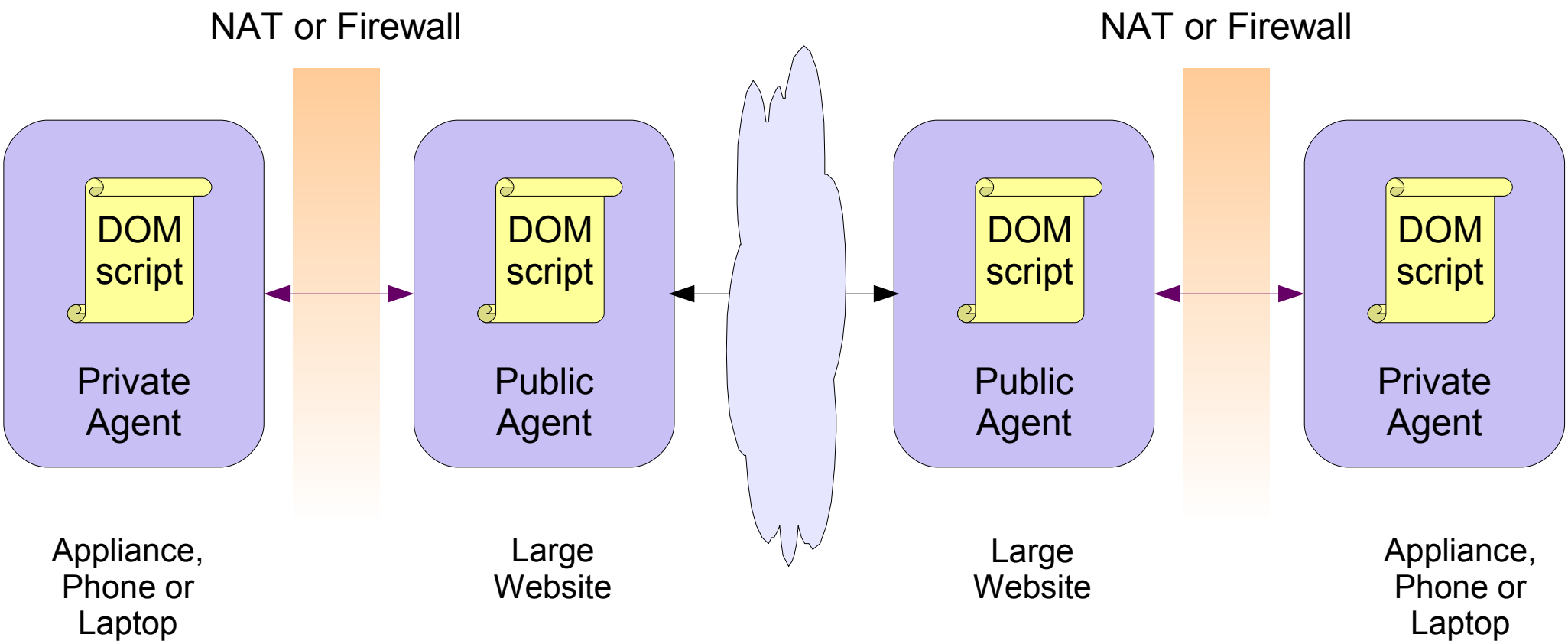
STUN and other techniques

Tunnelling through NAT



Connecting devices behind different NATs

Public and Private Agents



What's needed?

- Interfaces for accessing services from web scripts
  - Need standards for common services
  - Need standards for discovery and binding
- Descriptions that can be used for discovery and adaptation purposes
  - Semantic Web technologies like Ontologies
- Policies for discovery and binding
  - Need standards for describing them
  - Cover security and privacy considerations

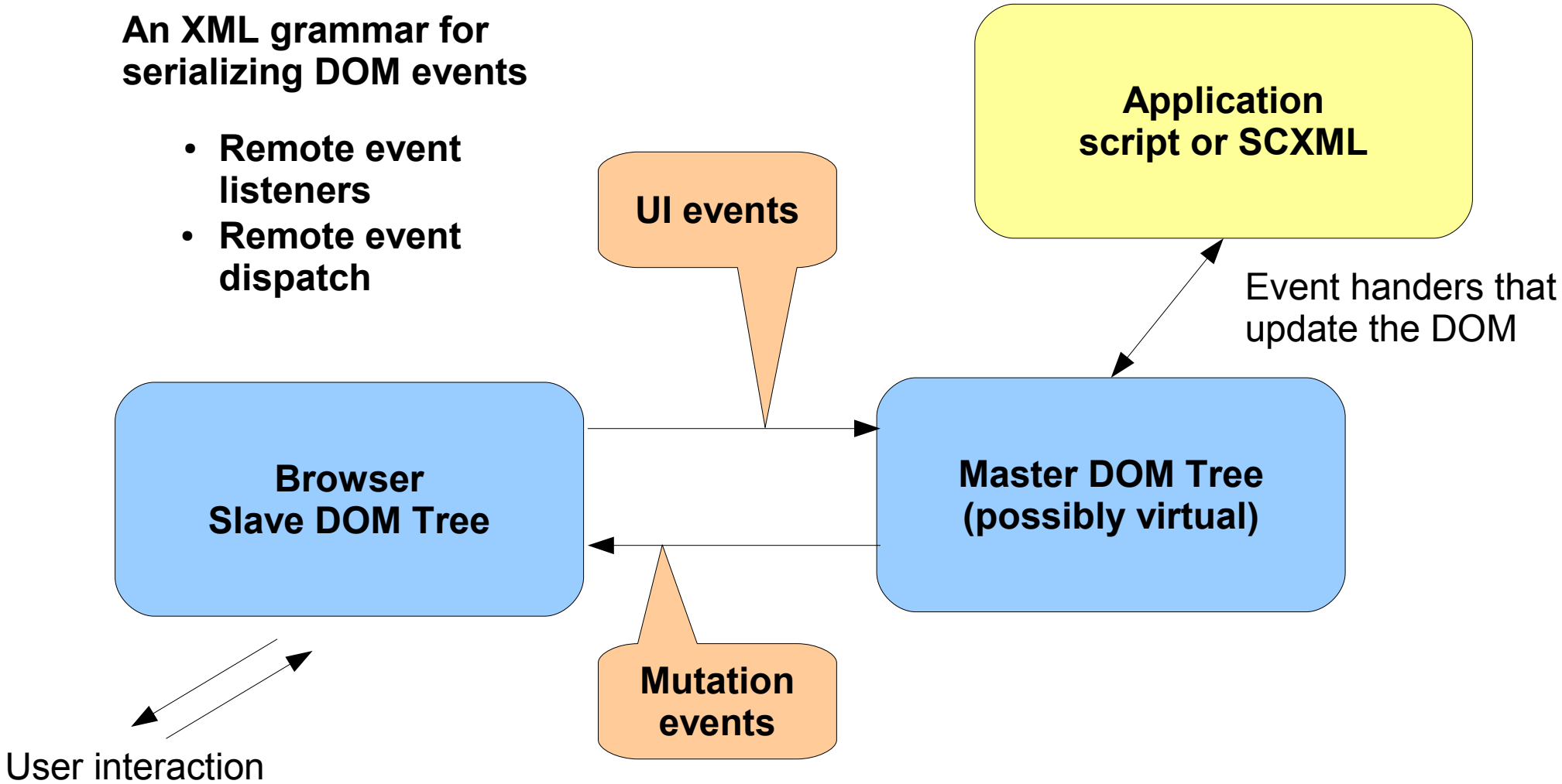
Remote User Interfaces

- Model behaviour as script or state machine
  - Interaction Manager (IM)
- Model UI as XML (XHTML, SVG, ...)
- Run UI and behaviour on separate devices
- IM sends events to update remote UI's DOM
- IM receives events from UI as result of user input
- UI can be distributed on multiple devices and controlled via single interaction manager
  - rich UI: mobile phone or remote + flat screen display
  - simple UI with buttons and indicator

Remote User interfaces

An XML grammar for serializing DOM events

- **Remote event listeners**
- **Remote event dispatch**

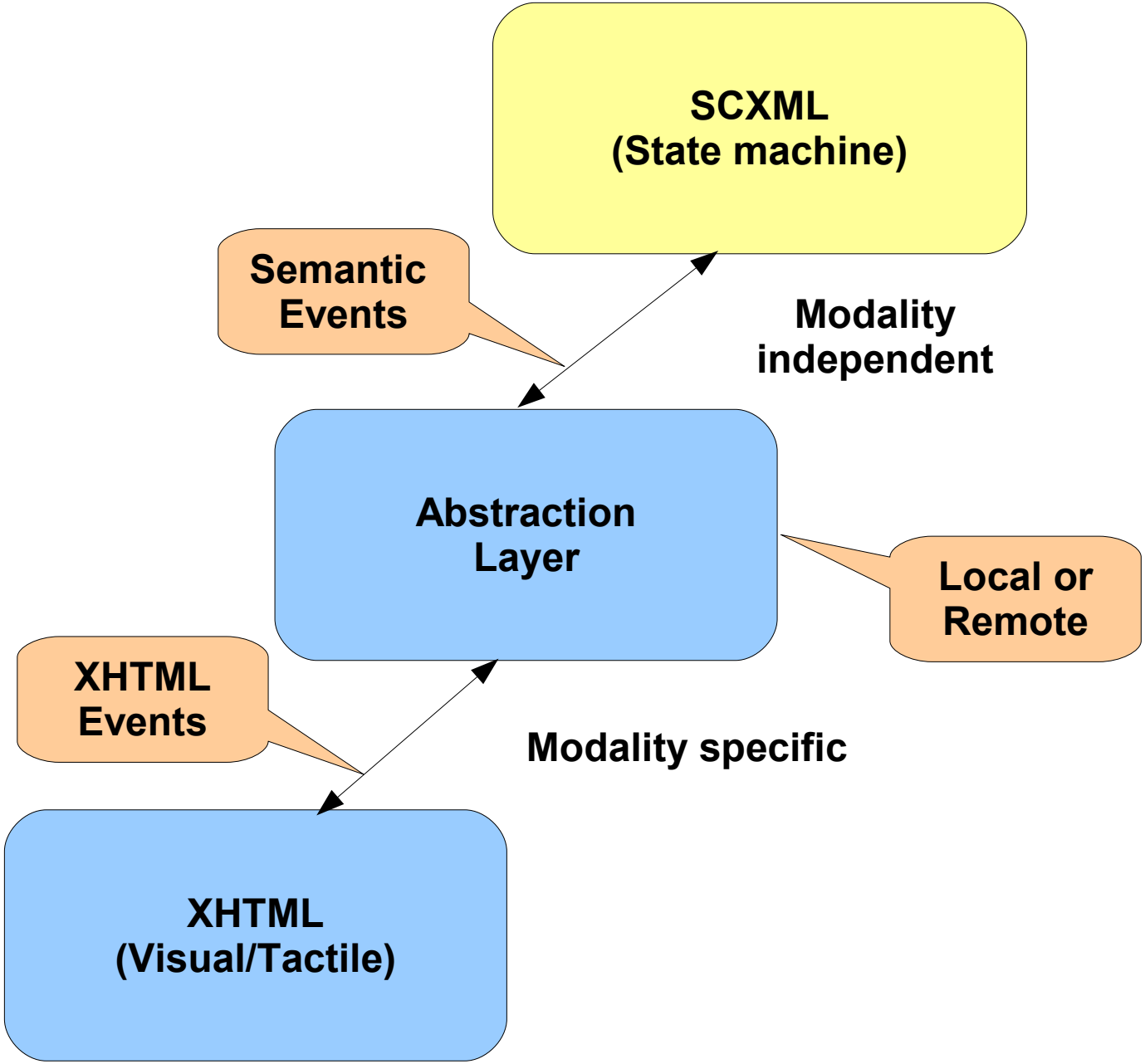


DOM = Document Object Model

Abstracting control

- Describe behaviour as event-driven state machine
  - Runs as agent
- Application level semantic events
- Couple UI to state machine via event transport
- XHTML + DOM operates at lower level of abstraction
- Introduce abstraction layer to mediate between XHTML events and application level events
- Abstraction layer can be located anywhere in network

Abstraction layer for Events



Adaptation

Describing applications in a way that makes them easier to run on a range of devices

Challenge of device diversity

- An ever increasing diversity of devices
- It is expensive to test on lots of devices
- My employer Volantis Systems has a database of over 4000 mobile devices with several hundred properties for each
  - browsers vary in details of scripting support, CSS bugs, etc.
  - variations in display size, fonts, kinds of buttons, memory, etc.
- Much tougher challenge than for desktop browsers

Policy-based Adaptation

- Author markup in device independent representation
  - authoring format is freed from browser restrictions
  - high level events in place of low level scripts
- Describe policies for adaptation to classes of devices
  - what layout, images, style sheets, scripts, etc.
  - skinning apps as combo of markup, CSS, script
- Adaptation process executes policies for specific delivery context
  - e.g. generate HTML4 if appropriate
  - split content for low memory devices
  - exploit client APIs for rich web apps (e.g. Ajax)

External Groups

with potential relevance to W3C work on Ubiquitous Web Apps

- 3GPP – protocols for mobile devices (GSM, W-CDMA)
- DLNA – device coordination for home entertainment
- FIPA – IEEE CS standards for agent-based technology
- HGI – devices acting as home gateways
- IETF – protocols including HTTP and SIP
- OMA – mobile application environment
- PUCC – device and service metadata for devices
- Others, e.g. UPnP Forum, JBMIA, PWG, OpenAjax Alliance

Ubiquitous Web Applications WG

- Home page http://www.w3.org/2007/uwa
- Follow on to former Device Independence WG
- Plus broadened focus on Ubiquitous Web Applications
- Looking for companies interested in working on
  - enabling applications across multiple devices
  - content adaptation for multi-channel delivery
- UWA WG Charter
  - http://www.w3.org/2006/10/uwa-charter.html
  - chair: Dave Raggett <dsr@w3.org>
  - team contact: Stéphane Boyera <boyera@w3.org>

UWA Workshop

- Dublin, Ireland, 5-6 June 2007
  - http://www.w3.org/2007/02/dmdwa-ws/
- Reduce the cost of developing and maintaining Web Applications
  - Capturing author's intentions
  - Abstract versus Concrete UI's
  - End to End models
  - Policies for adaptation to specific devices
  - Compositionality
  - Richer metadata for capabilities/services

Ubiquitous Web Applications

Questions?