The Ubiquitous Web

Dave Raggett (W3C/Volantis)

Web of Everything

- Incorporates a very wide range of devices
- Physical world of sensors and effectors
- Access to local and remote capabilities
- Autonomous agents that act upon behalf of their users

Ubiquitous Web

Exploiting Web technologies
to make it much easier to build
distributed multi-device applications

Web Mindset

- Markup

- Events

- Scripting

- URI's and HTTP

- Moving away from the client/server model

- Access to local and remote capabilities

- Hiding messy details of messaging protocols
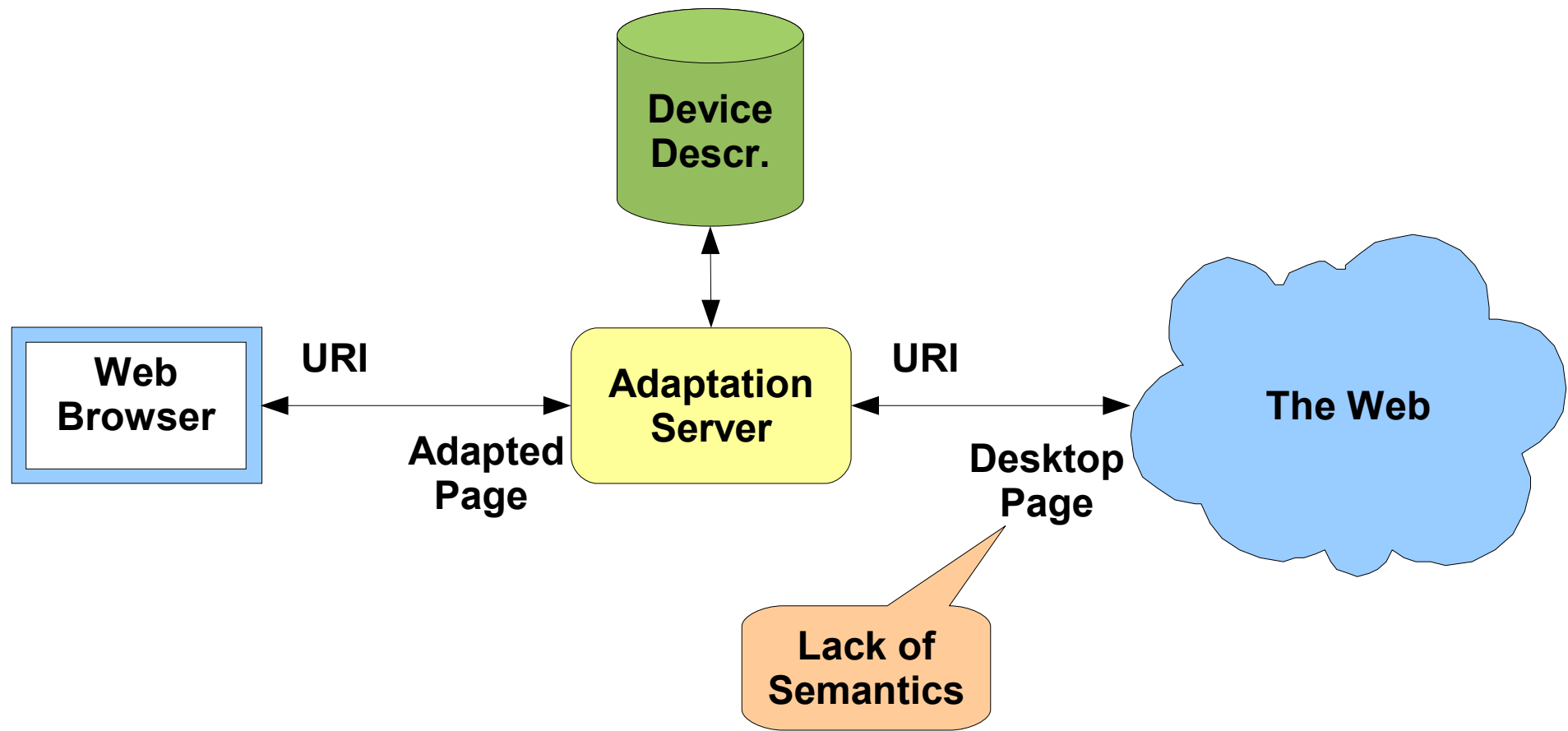
- Trust, identity, privacy and security

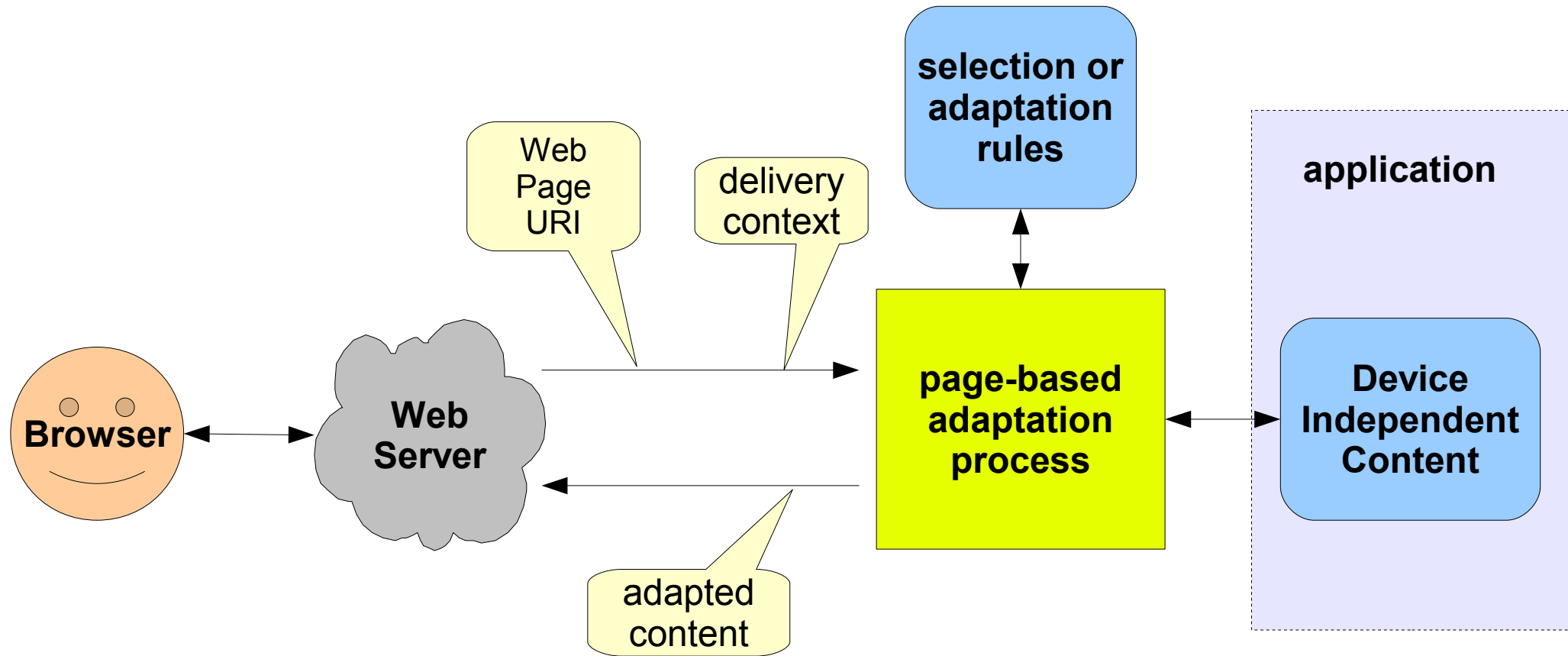# Increasing use of mobile devices

Increasing variety of devices

# Re-purposing Existing Content

Device Independent Content

DIAL – Device Independent Authoring Language (XHTML+XForms+CSDI)

Device Independence meets Ajax

- Strict separation of content and presentation make it easier to adapt to specific devices

- Selecting content based upon device characteristics to match end user needs

- Trend towards more interactive web pages

- Ajax and scripting for richer behavior
  - using HTTP from scripts without reloading page
  - dynamic manipulation of markup and styling
  - But client-side script acts on *transformed* markup

- Authors need declarative treatment of behavior

Higher Level Application Models

High level Application Model

delivery context

Policies
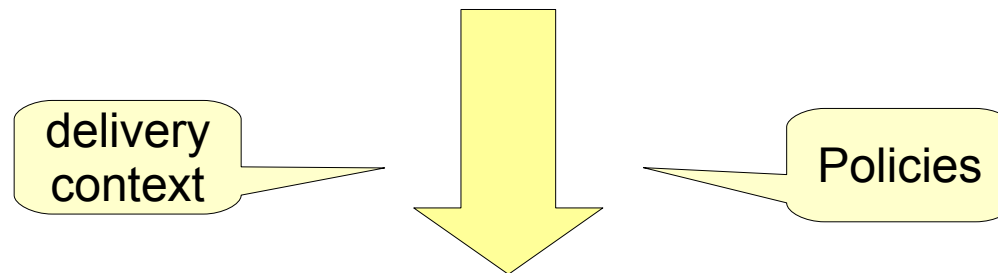
Device Dependent Application Model
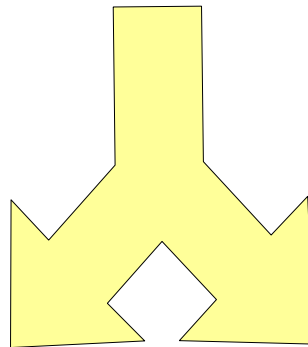
Client Components

Server Components

web pages

Interaction Manager

Policy based adaptation

- Interpret high level models in terms of lower level goals suitable for a given delivery context
  - Models describe application flow across pages
  - Goals dictate required content, behavior and styling
  - Combination of client and server side components
- Working around device/browser limitations
  - Recipes for modifying style sheets to work on particular devices
  - Recipes for assembling scripting modules to address desired behavior
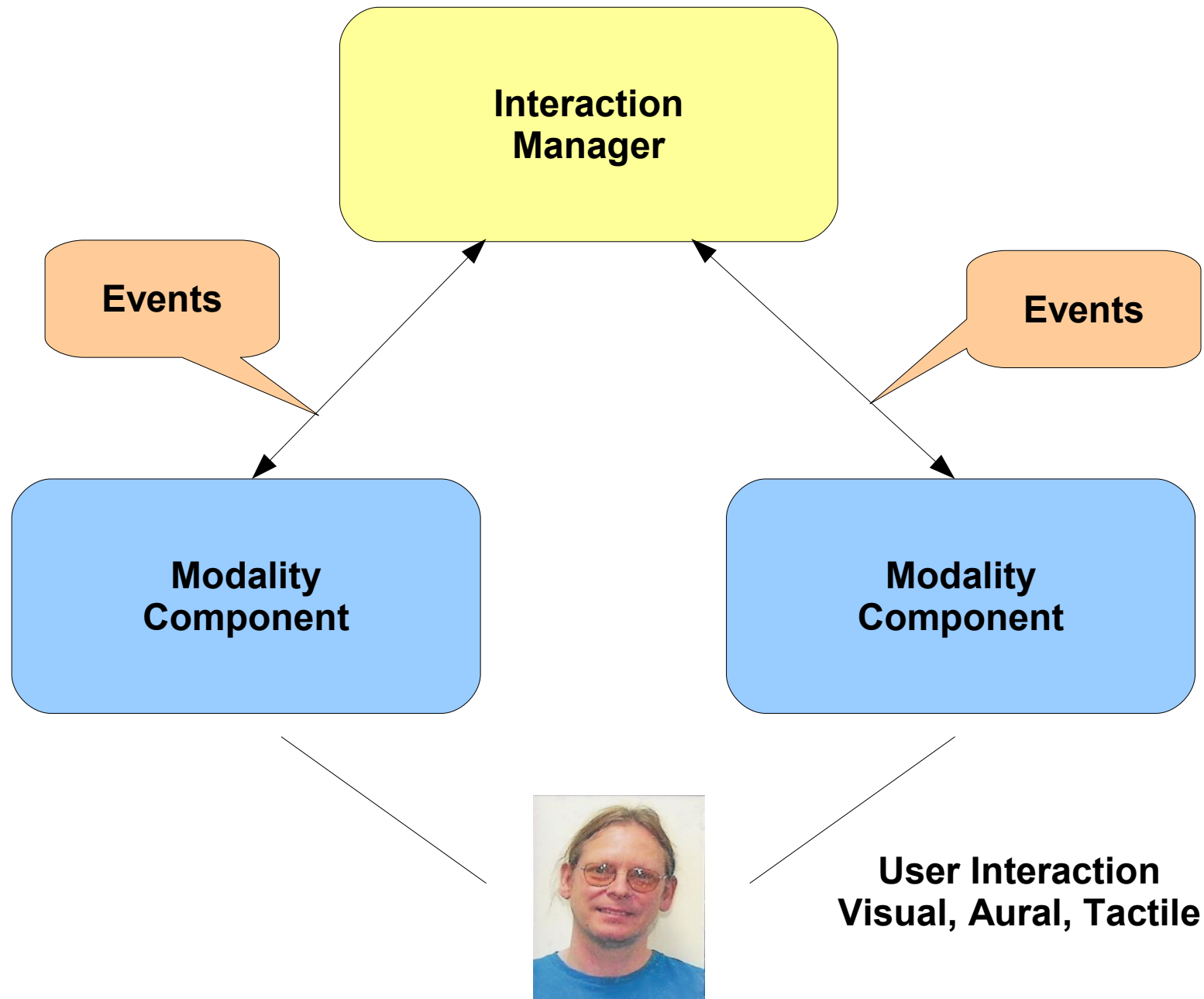
Declarative Web Application Models

- Plan to organize a W3C workshop in 2007
- Bring together practitioners to discuss use cases, requirements, solutions and research directions
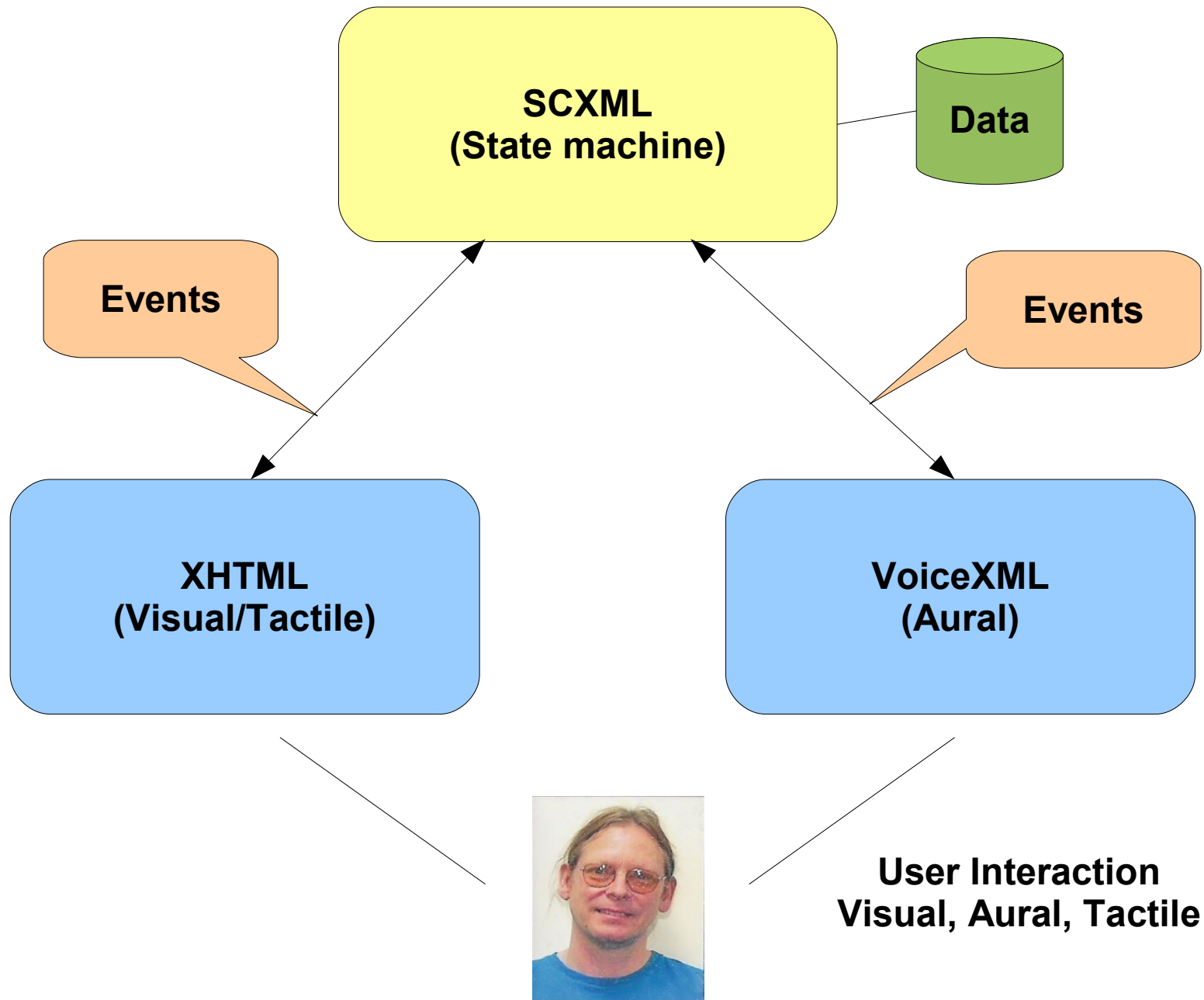- Location and date to be decided

Multimodal Interaction

Enabling people to choose how
they interact with Web applications

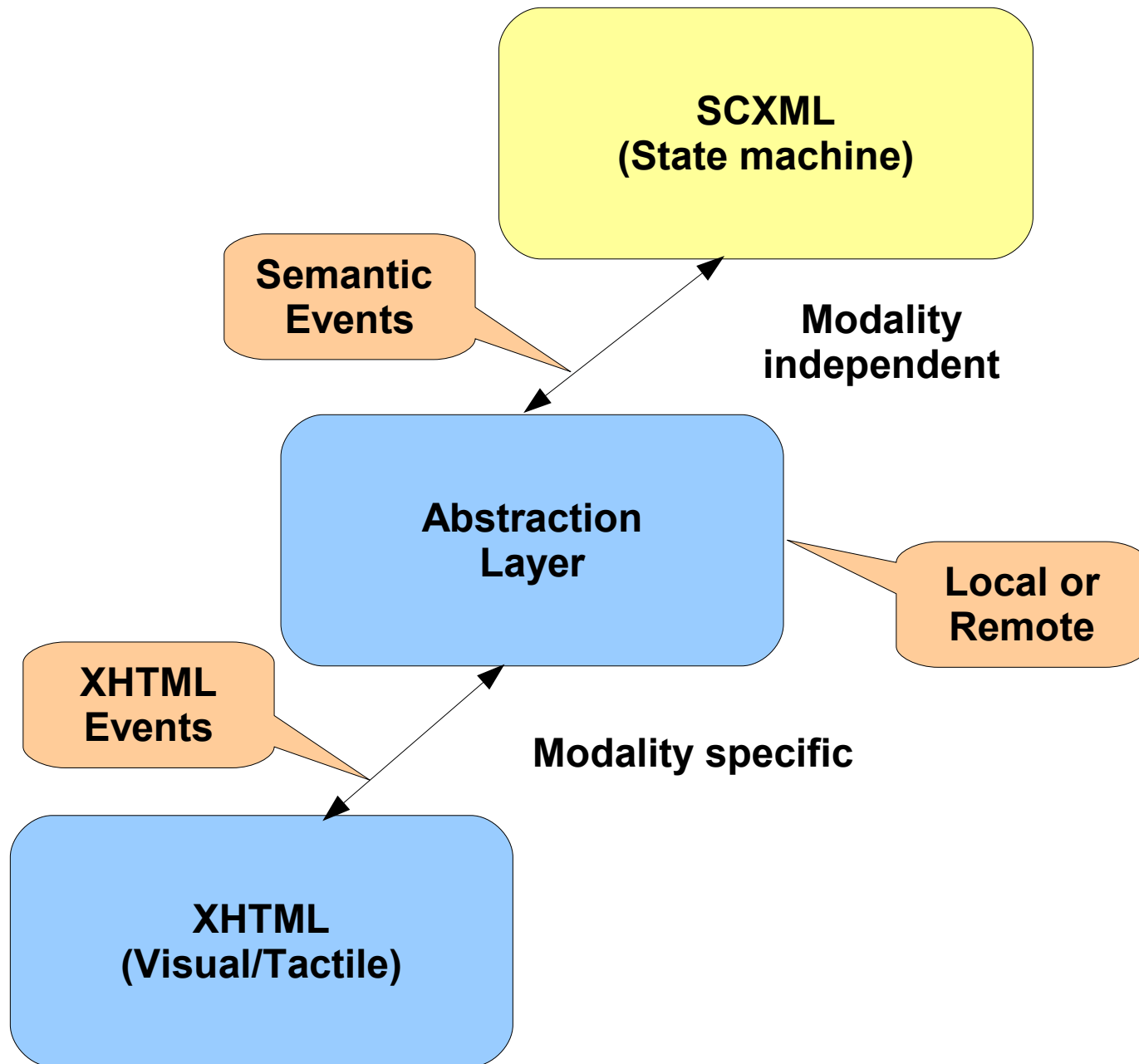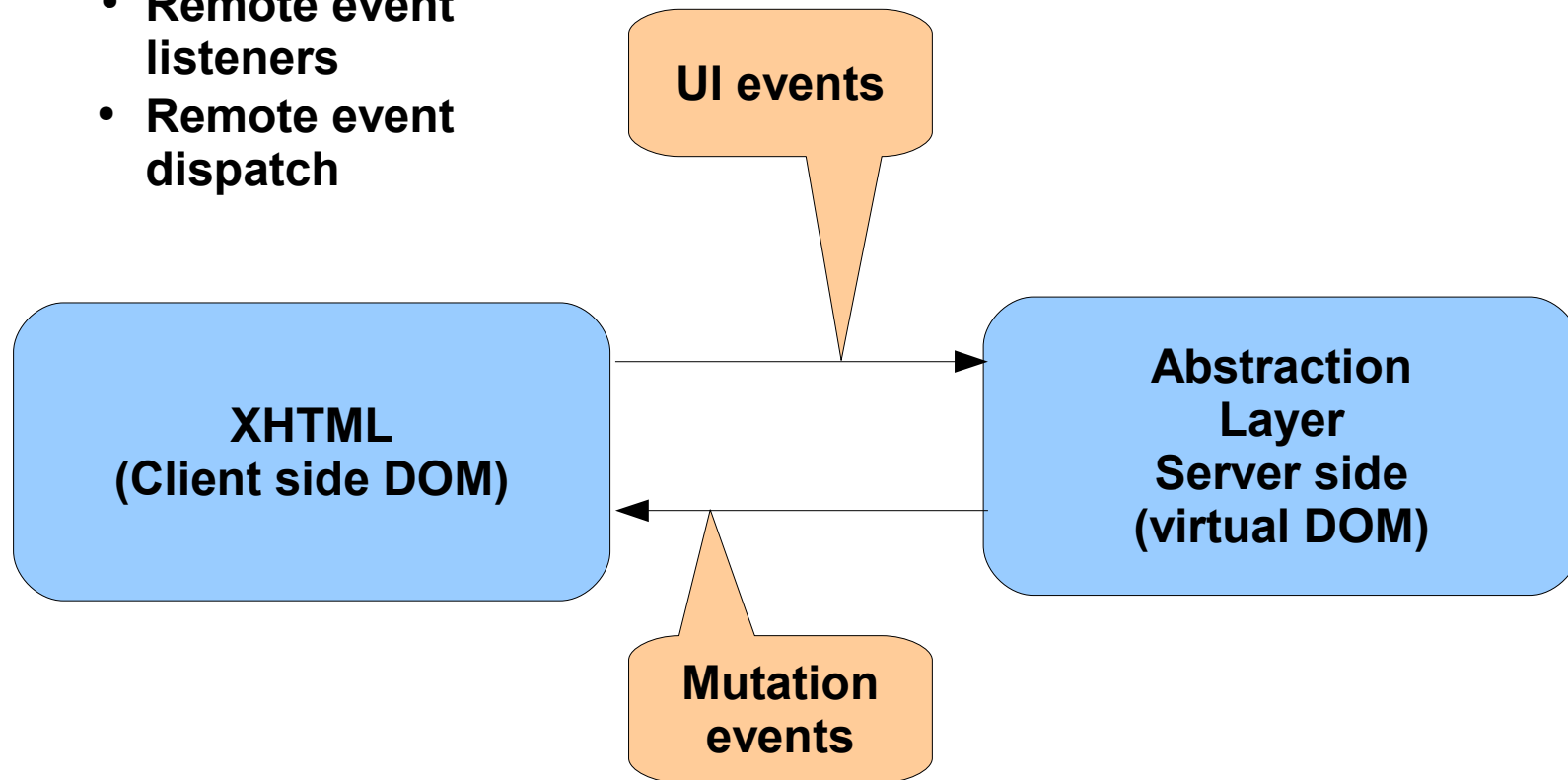# Multimodal Architecture

Multimodal Architecture

Abstraction layer for Events

REX for Distributed Components

REX – an XML grammar for serializing DOM events

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

UI events

XHTML (Client side DOM)

Abstraction Layer Server side (virtual DOM)

Mutation events

REX = Remote Events for XML
DOM = Document Object Model

Listening for external events

- HTTP - synchronous request/response pairs

- Allows web page to send events to server

- But how does the server send events to the web page?

- NAT and Firewalls block access to device

  – Outgoing connections okay, incoming are not

  – Problem for SIP, Web Services, etc.

- Possible workaround using HTTP and long lived connections

Is this a hack?

- Device sends HTTP POST Request with content type as application/rex+xml

- Server sends HTTP Response headers with content type as application/rex+xml

- Both parties then use HTTP chunked transfer encoding to send events over long lived connection as and when needed

- HTTP server acts as proxy for passing events between devices

- Works through NAT/Firewalls

Alternatives

- GSM networks and SMS
  - Phone-based applications register for SMS message types
  - Good for infrequent events where latency isn't a priority
- SIP events
  - Use of SIP to carry presence info over SIP
  - Part of 3GPP's IP Multimedia Subsystem (IMS)
  - Relies on proxies to operate through NAT/Firewalls
- Other specific standards, e.g. PictBridge

Web Agents

Evolving beyond the browser/website paradigm

DOM proxies for accessing services

DOM – XML Document Object Model
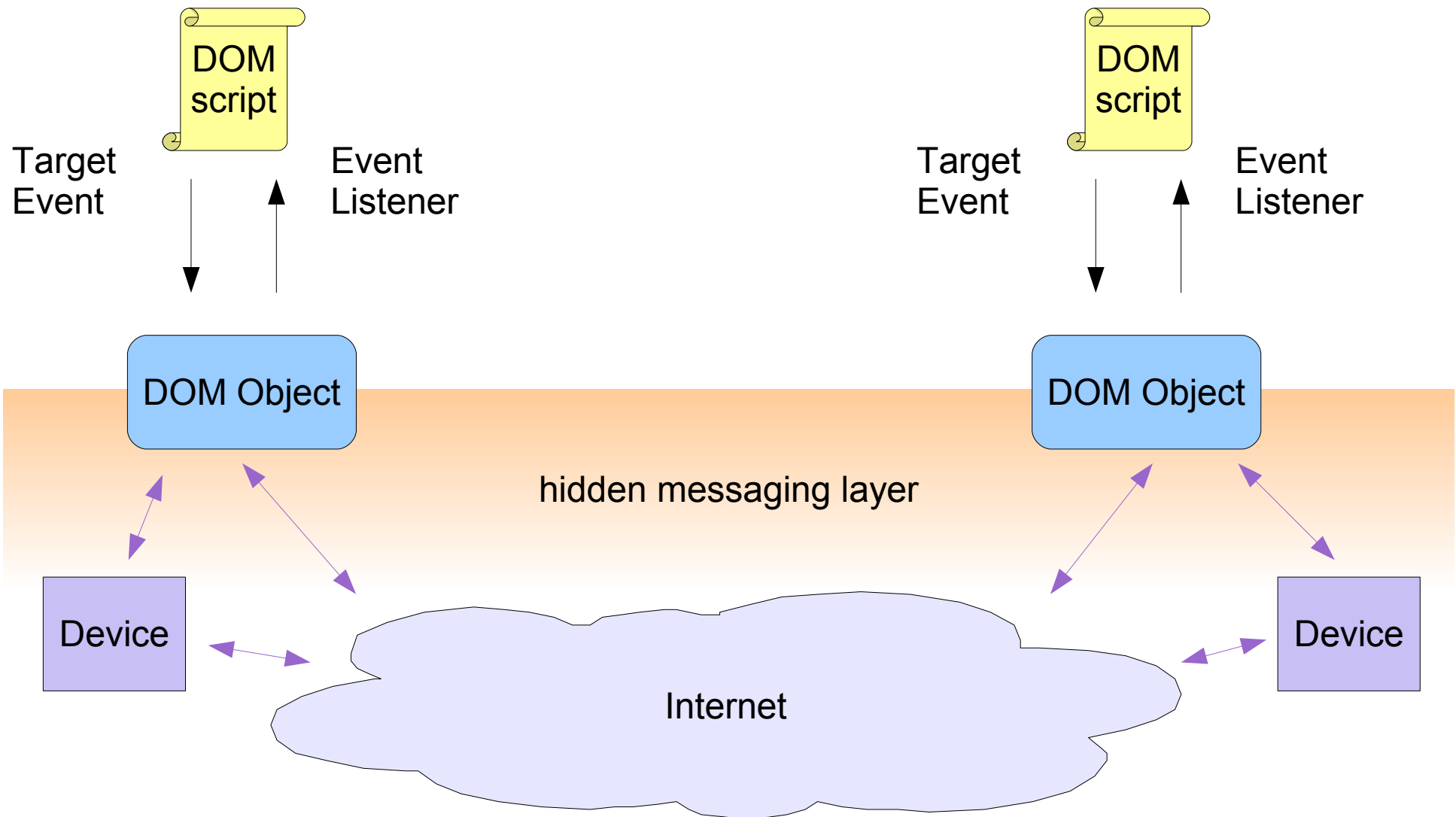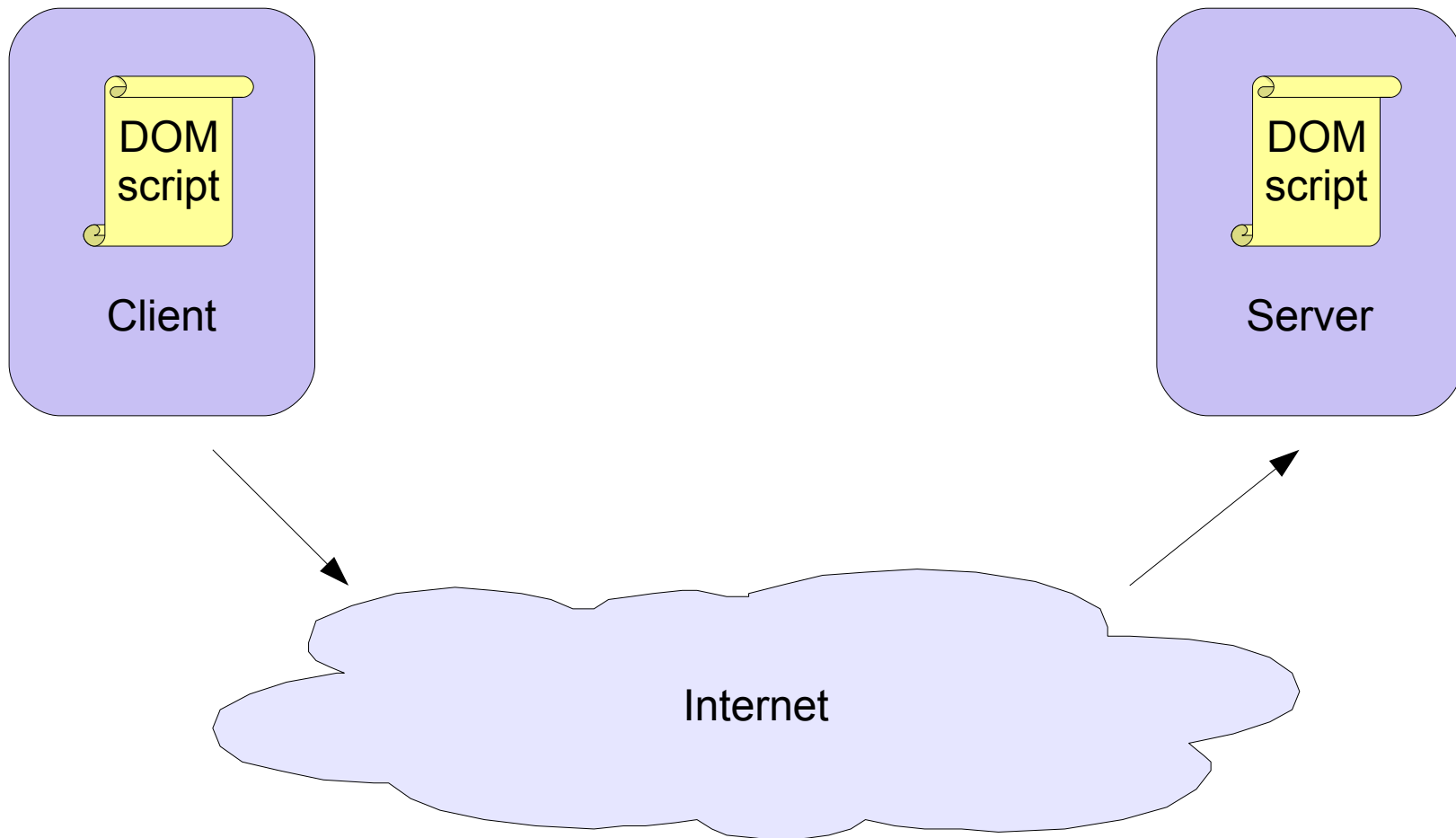
Client or Server?

Client or Server?

Agent combines client and server

Proxies

NAT or Firewall

DOM
script

Agent

Proxy

DOM
script

Agent

Internet

Public vs Private

NAT or Firewall

NAT or Firewall

DOM script

DOM script

DOM script

DOM script

Private Agent

Public Agent

Public Agent

Private Agent

Phone or Laptop

Large Website

Large Website

Phone or Laptop
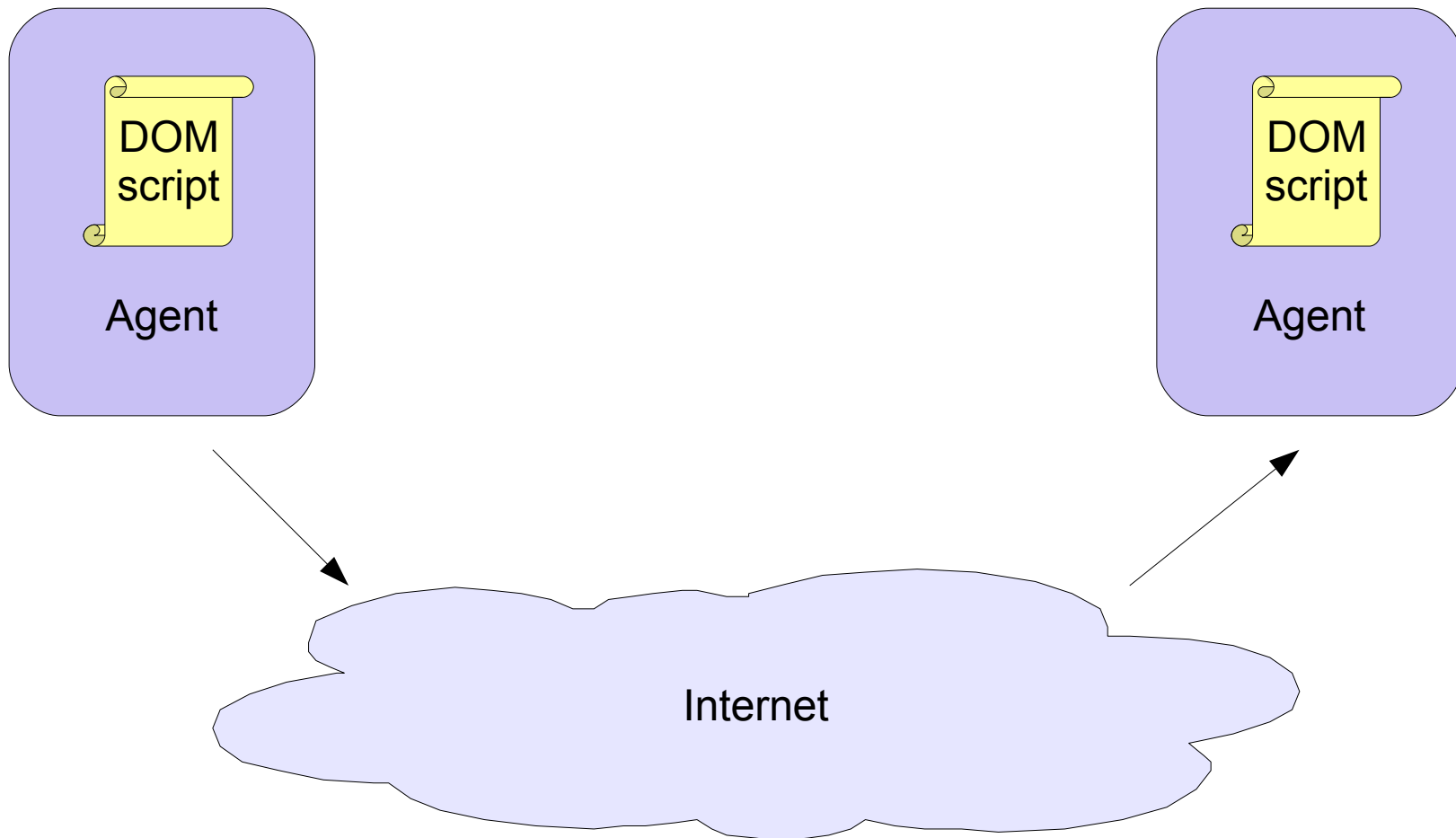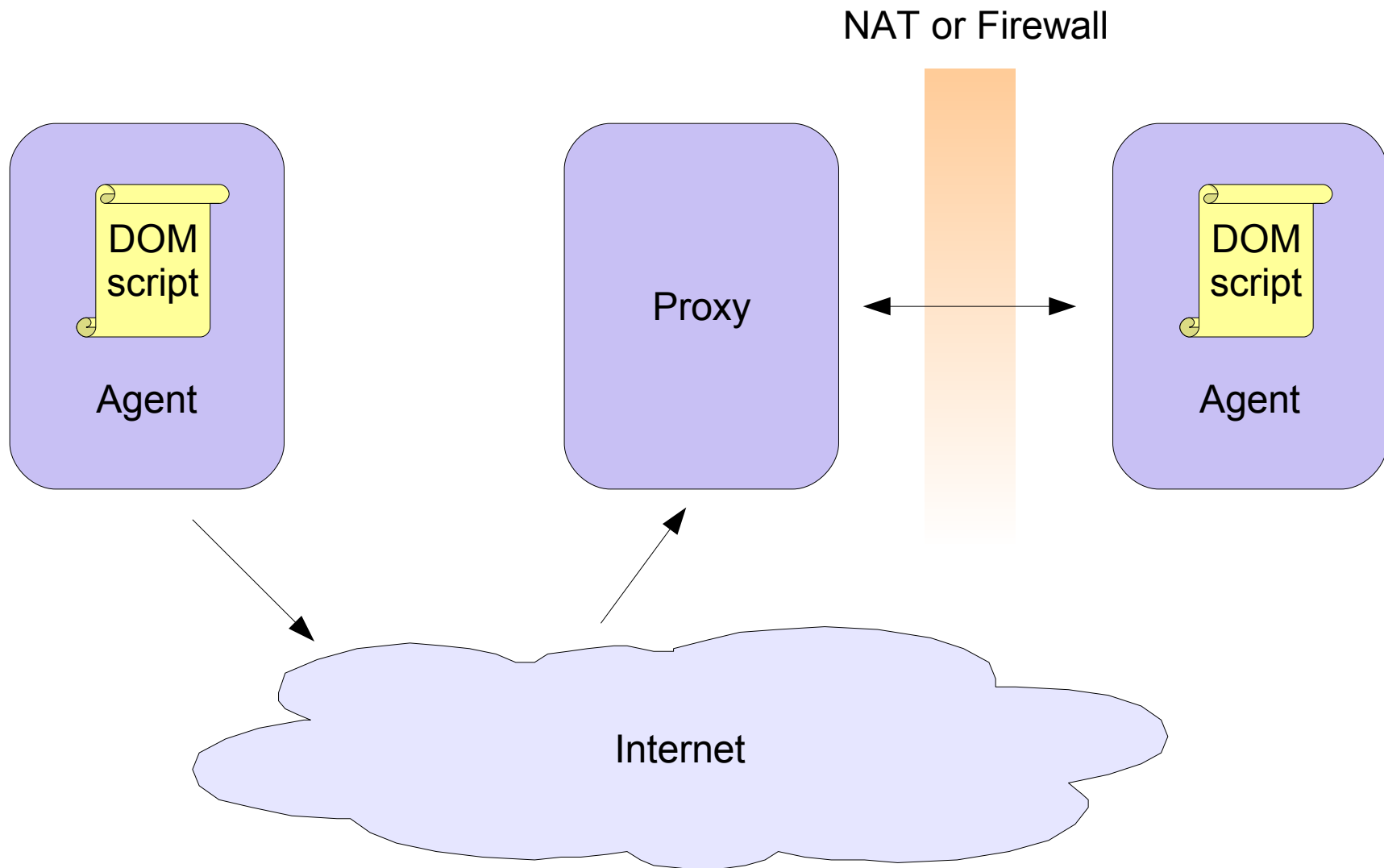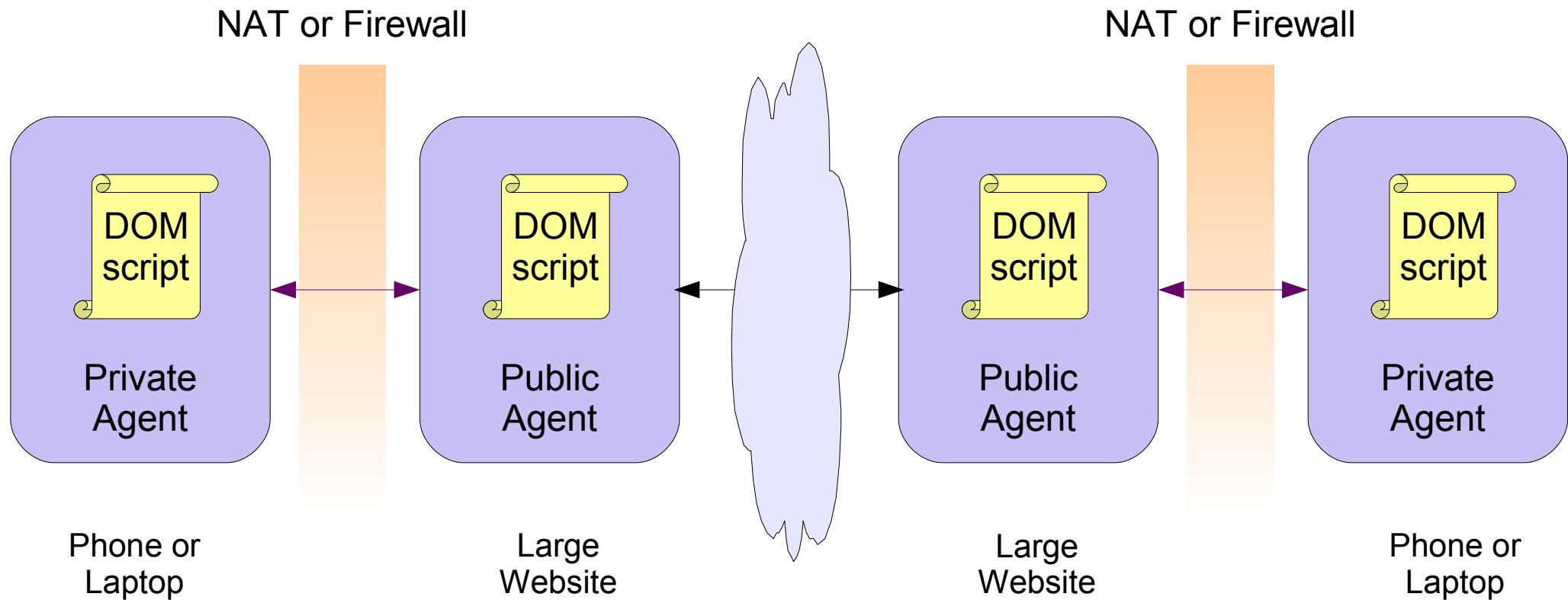
Autonomous Web Agents "WBOTS"

- Combination of markup and scripting

- Act on behalf of and controlled by users

- May run on mobile devices or large servers

- Provide particular services, e.g.

  - Downloading news overnight for viewing in morning

  - Providing your trusted friends or colleagues with information about your location and activity

  - Determining whether to let your phone ring or directing the call to voice mail depending on who is calling, time of day, your location, calendar etc.

WBOTS – pronounced as wuh-bots

Autonomous Web Agents "WBOTS"

- Wbots can provide services on behalf of users when the user is busy, sleeping, or just out of immediate contact

- You can't be "online" all the time, but your wbots can, and they never get tired!

- Users grant wbots with controlled access to device capabilities and external services
  - e.g. access to persistent storage, calendar, ...

- Users determine who can access services provided by their wbots

Trust, Identity, Privacy and Security

- Lots of problems
  - Having to remember a zillion passwords is a nightmare
  - Privacy – what privacy?
  - Fears of hackers, phishing and identity theft
- Ubiquitous Web applications will be dependent on finding better solutions
- Declarative techniques for trust and identity
  - SAML and Semantic Web
  - Federated Identity

Wbots in the home

- Providing access and control over
  - heating, lighting, and monitoring usage of utilities such as electricity, water, gas, internet, ...
- Tracking freshness of food in refrigerator
  - RFID for identity and sensors such as temperature
- Tracking where things are around the home
  - Mom, where did I leave the ...
- Media appliances
  - Music, video, photo's, television, ...

Plenty of other opportunities for use in cars, trains, planes, offices, factories, hospitals, schools

Supervisor wbots

- A special class of wbots that launch other wbots based upon particular events

- Launched when the device/server starts up

- Monitor behavior of other wbots

- Control access to services

- As an example, your phone might start the markup engine (formerly known as the "browser") along with a top-level supervisor wbot

- This listens for SMS messages and launches other bots as appropriate

Device Coordination

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

Service Discovery

- Name service or describe its characteristics
  - URI for service or service description
  - Description in 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 Web server
  - Based upon external description of service

Binding to a Service

- May be restricted and based upon proving membership of appropriate ACL

  - Issues of trust, identity, privacy and security

- Signalled by DOM event

- Binding in context of application session

- Session as service in its own right

  - Temporary sessions

  - Long lived sessions, e.g. meeting rooms

  - Access to "who" is participating in the session

  - Notifications when participants enter or leave

Binding in today's browsers

- HTML Object element
  - Naming a control via a MIME type
    - e.g. application/x-shockwave-flash
  - Naming a control via a GUID
    - Typically used on Windows
    - Relies on Windows Registry for version control
- Loading a named ActiveX object by a script
  - new ActiveXObject("Microsoft.XMLHTTP")
- Loading objects via XPConnect in Firefox
  - Components.classes[cid].createInstance()

Binding woes

- Current solutions generally specific to a given browser/platform

- Emergence of proprietary extensions
    - e.g. Opera Platform (for mobile devices)

- Lack of standards for common capabilities
    - e.g. to play/pause/resume/seek audio and video

- Ajax isn't the solution
    - Exposes pain of messaging protocols
    - Doesn't provide access to device capabilities

DCI

- W3C Framework intended for accessing user preferences, device capabilities and environmental factors

- Exposed as hierarchy of 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

DCI – Delivery Context Interface

What's needed?

- Interfaces for accessing services from web scripts
  - DCI is a relevant framework
  - 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

The Ubiquitous Web

Questions?