

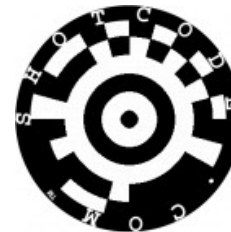
Ubiquitous Web Applications

# Towards the Web of Things

Dave Raggett, W3C & JustSystems

# Web of Things

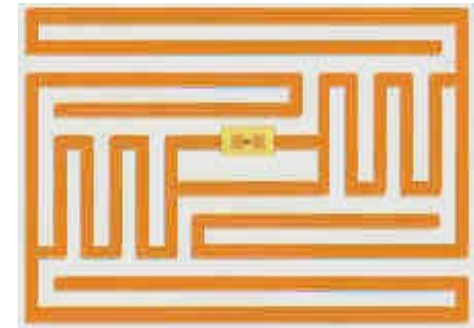
Barcodes as a way to connect physical objects to the Web



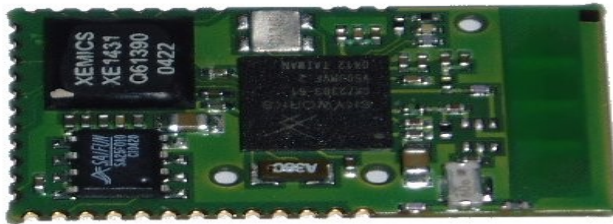
**Hyperlink your world!**  
With Semapedia you can connect Wikipedia knowledge with relevant places in physical space.  
[Learn more...](#)

# RFID

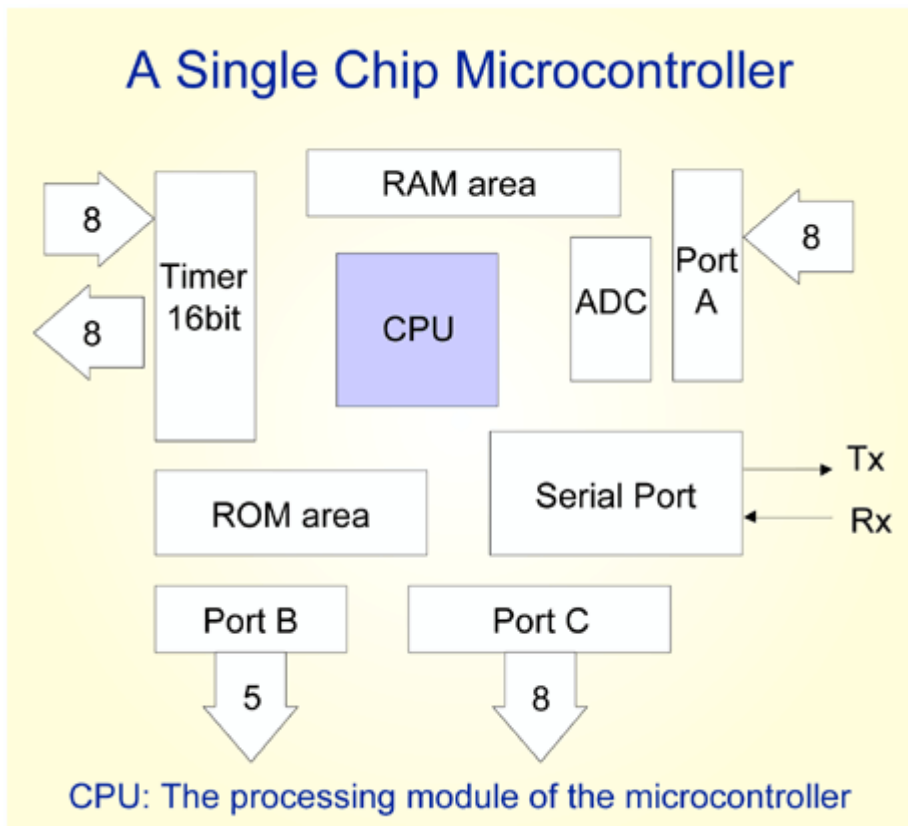
Electronic versions of barcodes  
but with extended capabilities



# Microcontrollers



- Computer on a chip
- Fastest growing segment of computer industry
- Average home now contains around 200
- Cars between 35 and 100 for luxury models
- Moore's law applies to networking circuitry



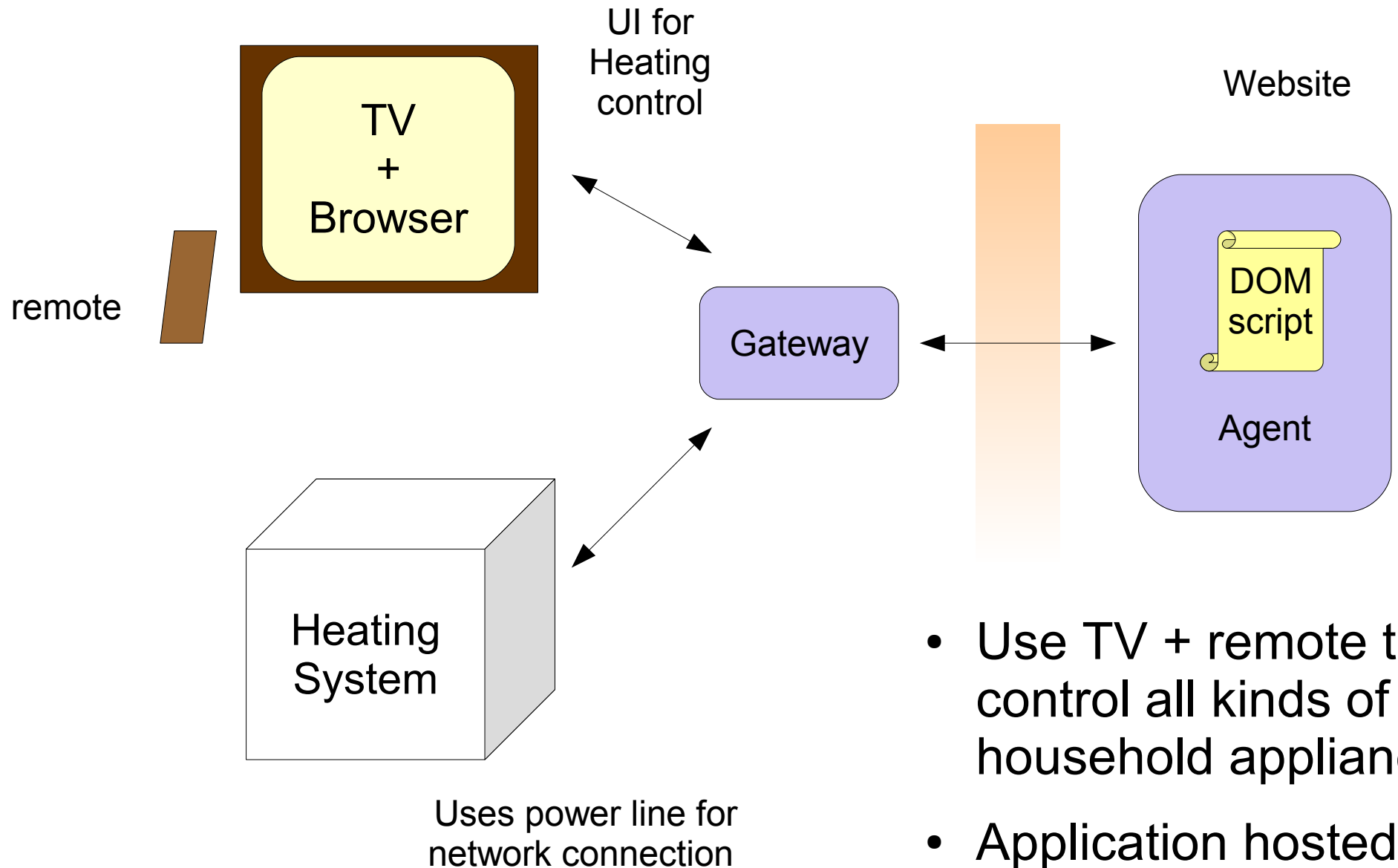
# Uses of Microcontrollers

- TV sets, TV remote controls, Video recorders printers, cameras, scanners, fax machines
- Ovens, toasters, refrigerators, washing machines, central heating systems
- Mobile phones, PDAs, MP3 players, computer monitors
- Car body electronics, air conditioning, seat control, chassis and safety, infotainment, power train
- The list goes on and on ...

# Web of Things

- Rapidly diminishing incremental cost for networking all kinds of devices
- The challenge for how to integrate devices as part of distributed applications
- Changing the way we think of the Web
  - No longer just about viewing websites on desktop browsers with big screens
  - Instead apply Web technologies to ease the task of developing new kinds of applications across a very wide range of devices

# Home network example



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

# Home environment example

- You are returning from a week's skiing vacation
- You are waiting in the airport for the flight home and think how cold your apartment will be
- You pull out your phone and open the browser to the bookmark for your home environment
- You select the temperature section and set the thermostat and timer to warm the apartment nicely by the time you expect to get home



# Remote printing example

- Imagine you are out with friends and taking photos with your phone
- You open the browser and select some of the photos that you want to print on your home printer
- The Web application talks to the printer and determines whether it has enough ink and paper
- Your photos are ready and waiting when you get home

# Home security example

- Your phone vibrates to alert you to a new priority message
- It is from the company you are using to look after your home security
- You click on the link to access the cameras in your home
- Some kids have kicked a ball through one of the windows
- You click on another link to call the building manager to arrange to have the window fixed

# Car navigation example

- George is driving his SUV, a 2010 Toyota Highlander to work in Los Angeles
  - He doesn't program the navigation system since he does this journey every day
  - But the car has learned the destination from his daily habits
- The navigation system advises him to turn left instead of right, warning him of the hold up from an accident on his usual route
- This service is provided by “Freestyle office”, a new service from Toyota and AT&T
  - This includes voice based email, news and a calendar system allowing George to prepare for the coming day

# 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

# Business Challenges

- Increased global competition is squeezing profit margins on consumer electronics
- Higher volume production runs can reduce costs, but this has limited effectiveness
- Companies need to look to new ways to create value for their products
- The means to link devices together creates new business opportunities
- This requires cross company collaborations and strong standards to build consumer confidence

# Business Opportunities

*through value-added services*

- Profitable services with clear value proposition for users
- Innovating with ways to supplement low profit margins on devices
- Building upon experience with mobile business models
  - Service contracts
  - Pay as you go
  - Ad supported services
  - Encouraging people to upgrade

# 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

# Ubiquitous Web Applications

## Architectural Challenges



# Networking Technologies

- Applications will need to work over a mix of rapidly evolving networking technologies
  - Ethernet over twisted pair or coax
  - DSL over copper phone lines
  - Ethernet over building power wiring
  - WiFi and WiMax
  - Bluetooth
  - ZigBee sensor networks
  - Near field communications
  - GSM and cellular packet radio

# Coping with Change

- Devices are continuing to evolve rapidly
- Some devices are in use for many years
  - televisions, heating systems
- Others are replaced quite frequently
  - mobile phones upgraded every 18 months (or so)
- Coping with minor malfunctions in ageing kit

# Coping with Change

- Developers need ways to create applications that can
  - cope with a mix of device vendors
  - cope with a mix of device generations
    - and likelihood of new versions of software APIs
  - cope with a mix of networking technologies
  - cope with minor device malfunctions
- Solve through mix of standards and modular architecture that minimizes dependencies

# Ubiquitous Web Applications

## Device Coordination

# Device Coordination

- The means for devices to expose their capabilities/services
  - rich descriptions and APIs
- The means to search for and bind to such services whether local or remote
  - brokers, security and trust management
- The means to exchange events across devices and services
  - asynchronous communications
- The means to coordinate the operation of one or more devices (managed services)

# Rich Descriptions

- Apply Web technologies for rich descriptions
  - ontology as meta model (data about models)
  - ontologies that describe data models, service models, trust models, and relationships
- Enabling applications to dynamically adapt to the changing context
  - user preferences, device capabilities and environmental conditions
  - descriptions of APIs and versioning
- Reasoning over security policies and trust relationships

# Device Ontologies and APIs

- Current focus on mobile devices, but other kinds of devices are expected to follow
- Risk of market fragmentation as each company define its own API for accessing device capabilities
- Increasing importance of defining common standards with involvement of all stakeholders
  - first tackle simple properties e.g. screen orientation, volume level, vibrator on/off, battery level, etc.
  - later tackle harder properties e.g. location
    - security and trust implications, legal framework

# Ambient Intelligence

Dynamically adapting to the Delivery Context

- User
- Device
- Environment



# User Preferences

- Some people prefer tiny fonts, while others can only read text in big fonts
- Some people require high contrast and may be unable to distinguish certain colours
- Some people are more sensitive to the price they are paying for data and want smaller pages
- Some people may be willing to see advertisements if this means content is free

# Device Variations

- Variations across browsers
  - markup, scripting, style sheets, media support
  - very expensive for design and testing
- Variations in screen size
  - major impact on usability
- Variations in available memory
  - may be unable to load large web pages
- Huge gap in capabilities between high-end smart phones and the rest
- Need to support heterogeneous mix of devices

# Environmental Factors

- **Bandwidth**
  - not everyone is on a high speed connection
  - applications that adapt to changing bandwidth
- **Web applications that can work offline**
  - and sync up when next connected
- **Battery Level**
  - large pages with big scripts drain the battery
- **Location**
  - huge potential for location based services

# Delivery Context

- Descriptions of user preferences, device capabilities and environmental conditions
  - For individual users and devices
  - For classes of devices, e.g. all Nokia N95s
- Exposed through APIs
  - Client and Server-side
- Delivery Context Ontology
  - defines concepts and relationships
  - provides underlying model for APIs

# Content Adaptation

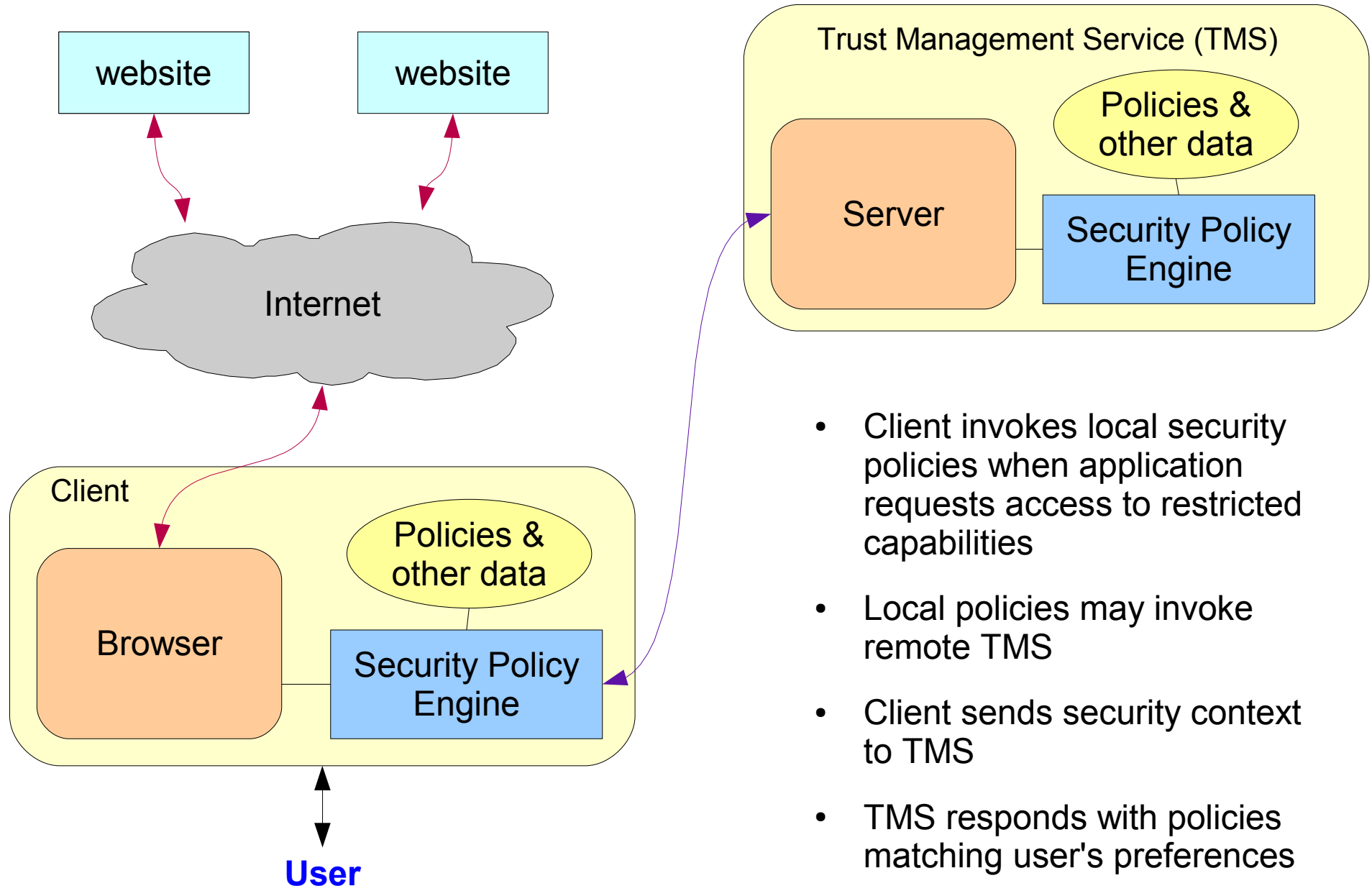
Through access to the Delivery Context

- Authoring time
  - Design for different classes of devices
- Request time
  - Taking details of a specific device into account when a page is requested by an HTTP client
- Run time
  - Dynamic adaptation after page has loaded
  - Dynamic adaptation of media streams

# 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



- Client invokes local security policies when application requests access to restricted capabilities
- Local policies may invoke remote TMS
- Client sends security context to TMS
- TMS responds with policies matching user's preferences

# The Web of Things

- The Ubiquitous Web will involve pretty much all networked devices
  - Mobile devices are just the start
- Next step is to look at range of consumer electronics
  - printers, cameras, media servers, digital TVs
  - making these devices into first class web devices
- Using XML to specify device behavior
  - Loose coupling of devices through events
  - Layered architecture for high level authoring



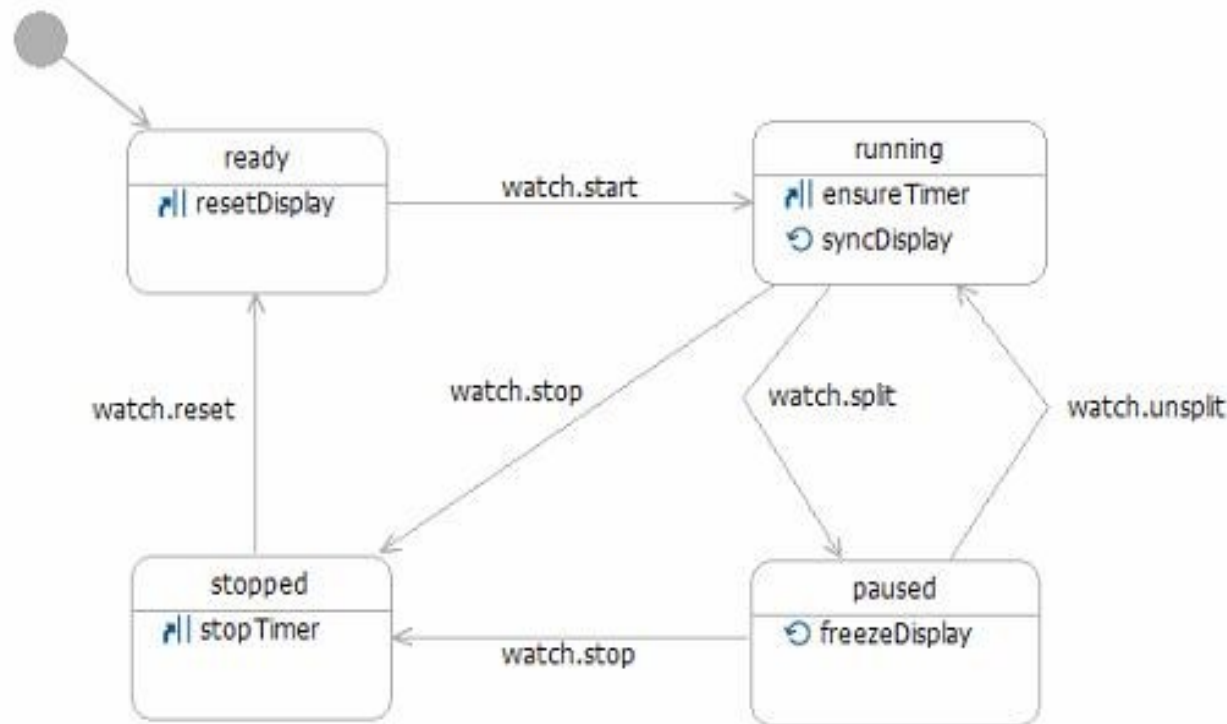
# Eventing

- Used to couple devices and services as part of distributed applications
- XML based model of device behavior with a document object model (DOM)
- Application developer can set event handlers and can target events at DOM objects
- When the application wants to make use of some other device/resource it binds that as a proxy object in the local DOM
- Proxy object hides the details of communication with resource

# Executable Models of Behavior

XML plus diagrams for ease of authoring

- Models of tasks and the data they operate on
- State Chart XML for event driven state transition models



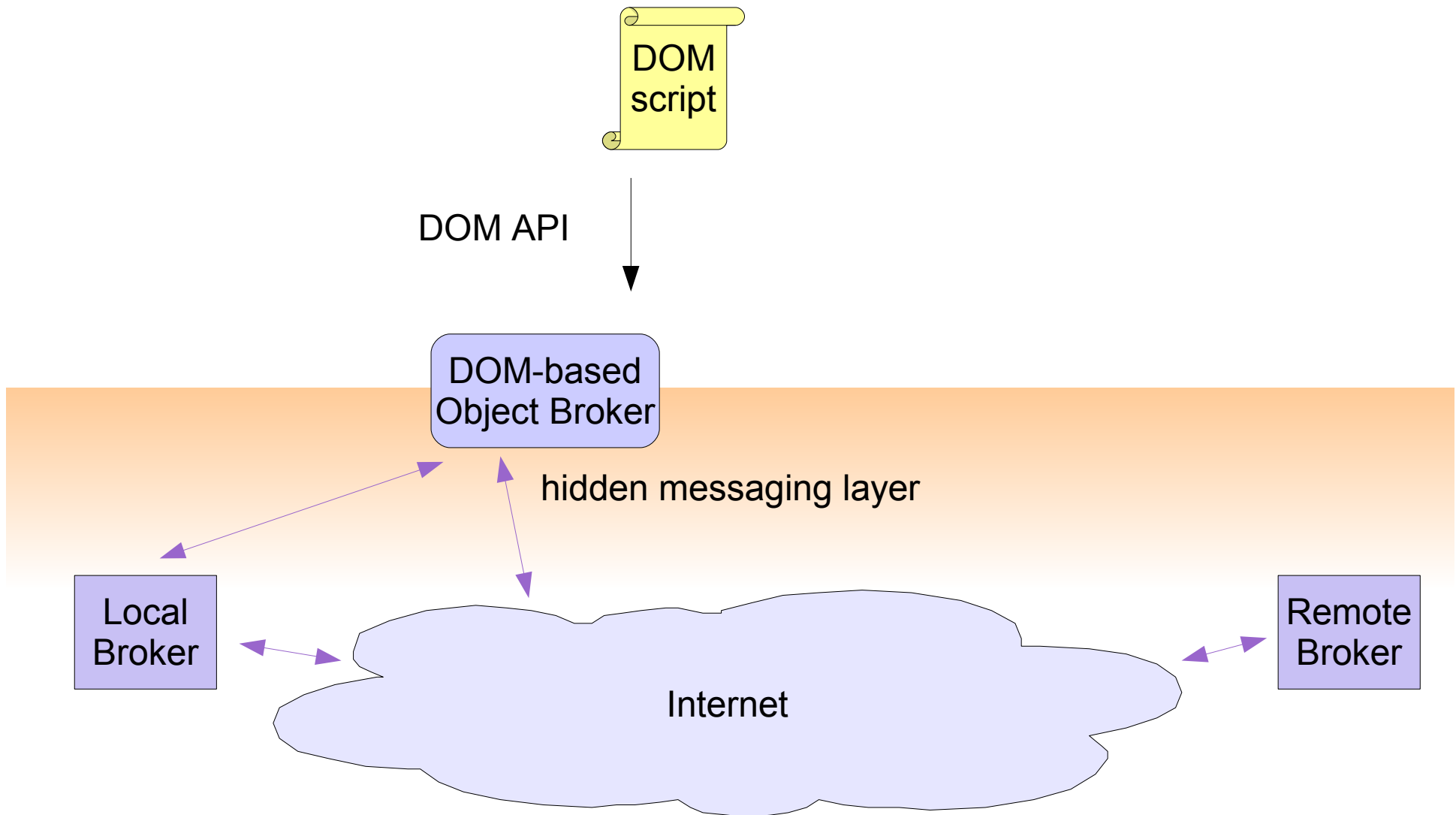
# Declarative Models of Behavior

- Some devices have a fixed function and behavior
- Others are programmable via download
- State charts and rules instead of Java and C++
  - Plus libraries of predefined objects
- Can be delivered as XML and interpreted
- Or compiled to bytecodes for Virtual Machine
  - For reduced memory/processing needs

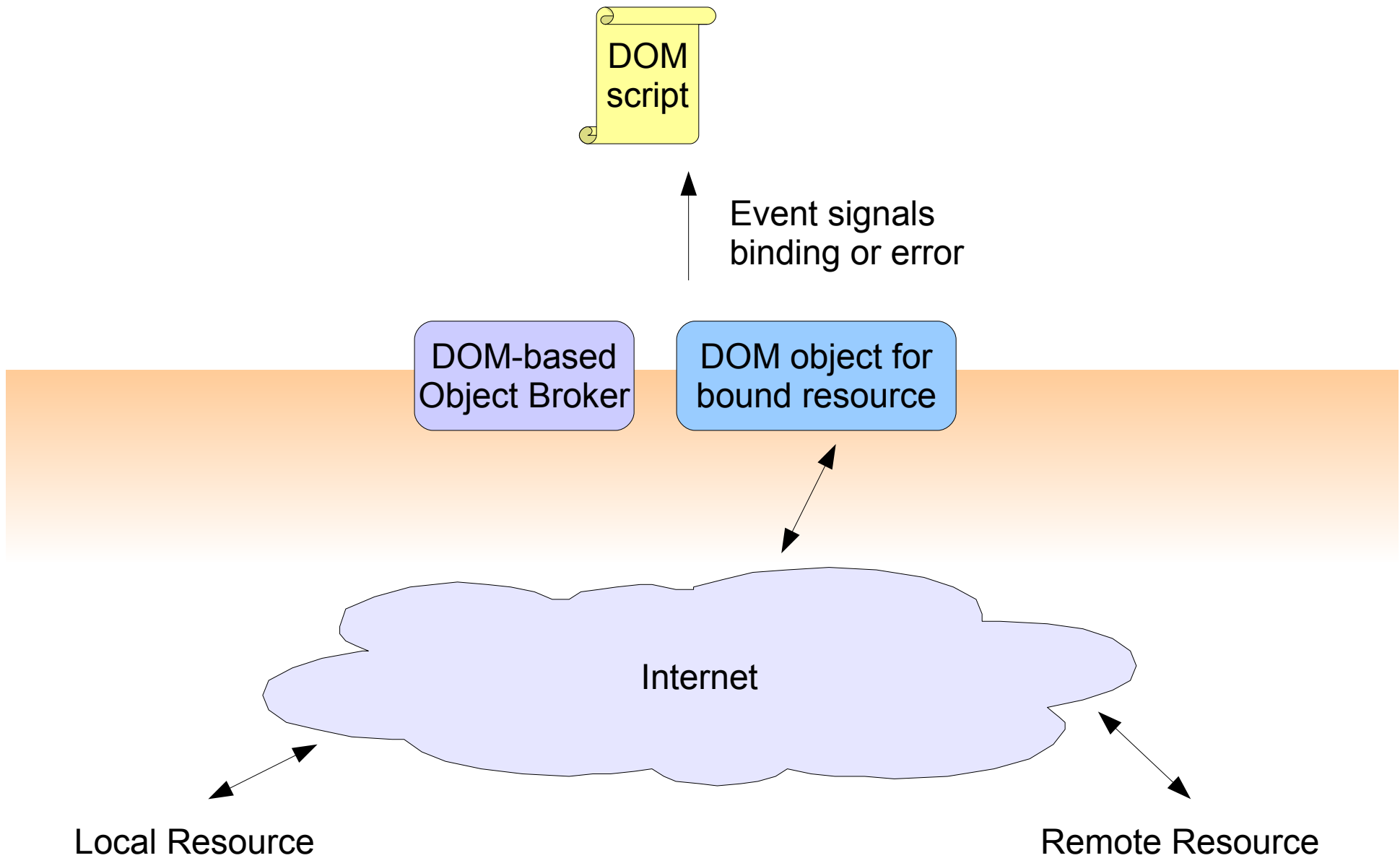
# Device Abstraction Layer

- Web applications access device services through a device abstraction layer exposed as objects in the browser DOM
  - the Web as a kind of ubiquitous operating system
- This hides the low level device API and associated communication protocols
  - an overlay for heterogeneous environments
- Applications can set event handlers, and can target events to invoke services
- Events can carry complex data using an XML-based data model

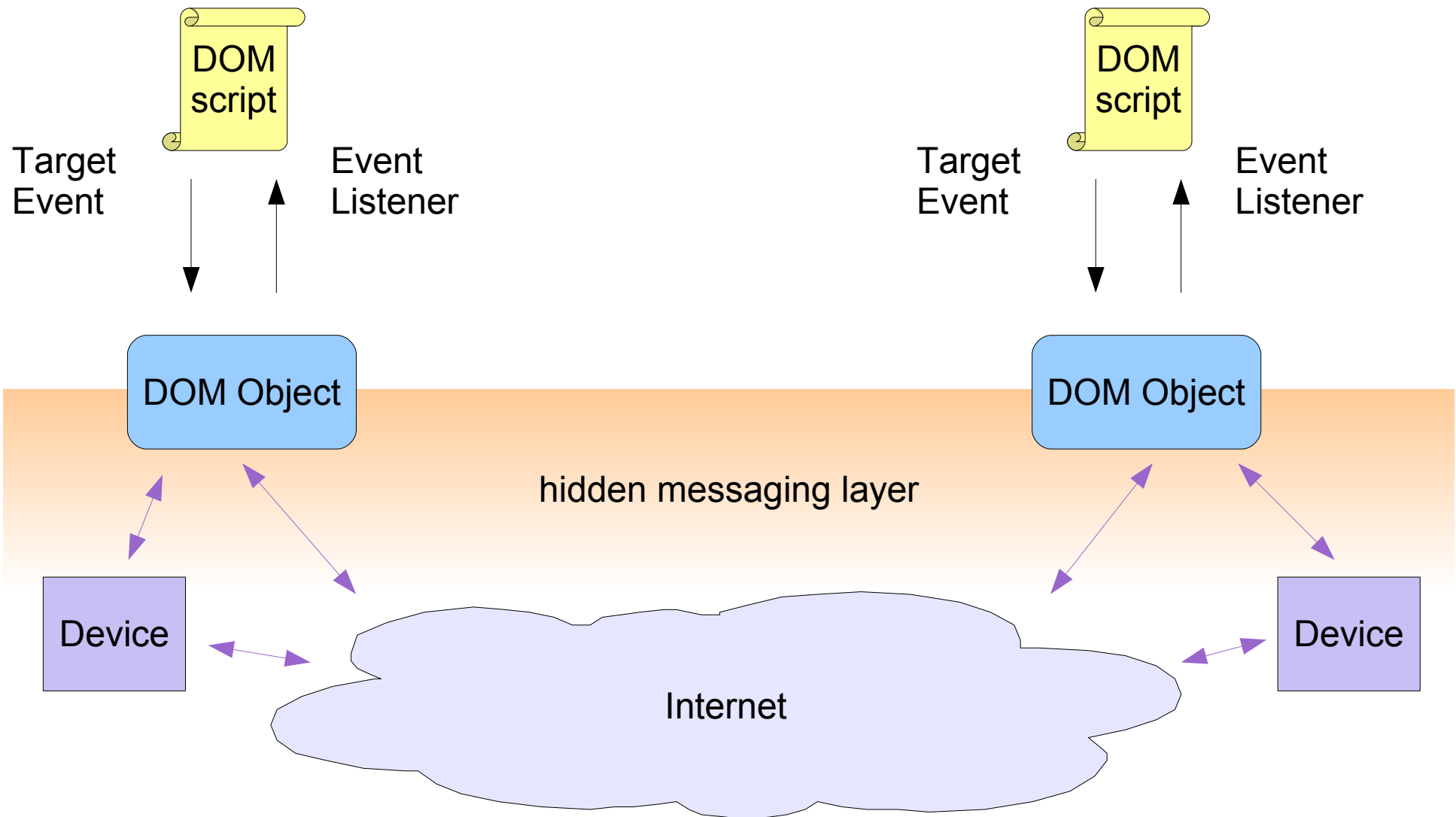
# Proxies for accessing brokers



# Proxies for accessing brokers



# Proxies for accessing services

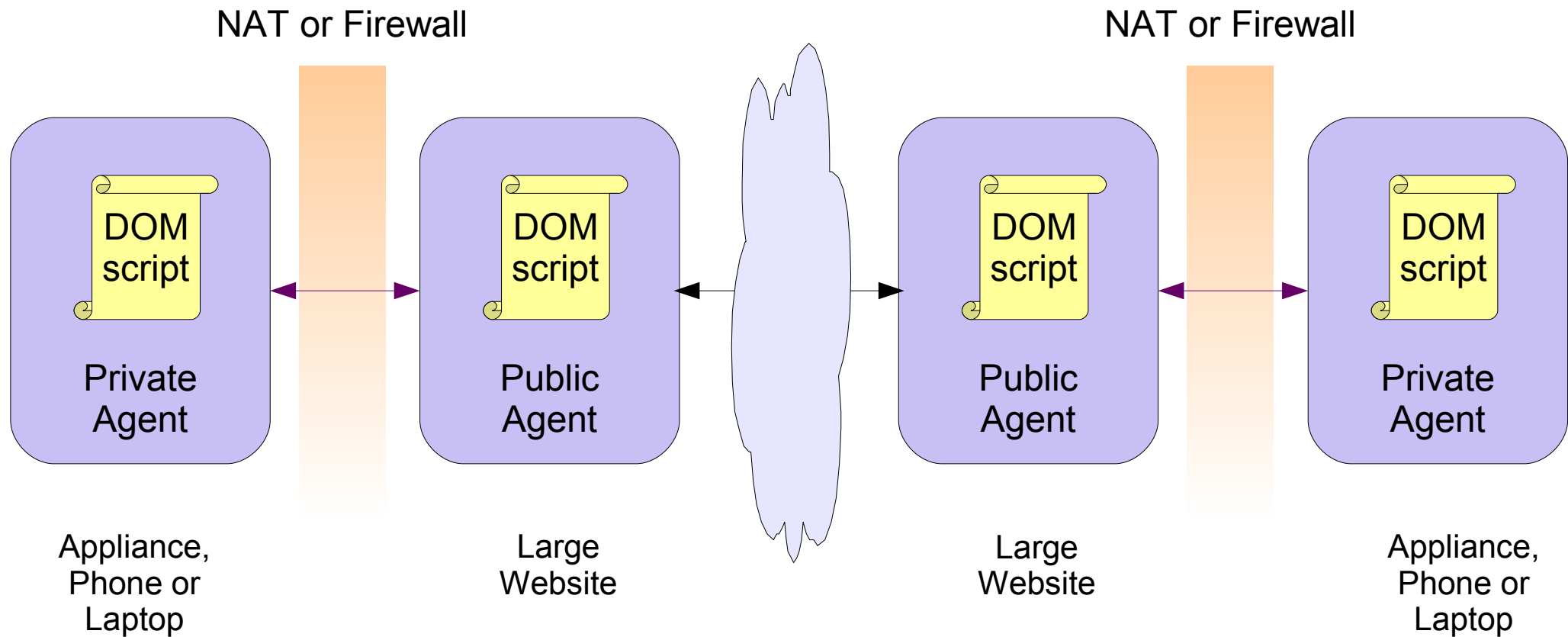


# Agents

- Web-based applications that run on local or remote devices (e.g. large websites)
- Listen for incoming requests and in turn send requests to other agents
- Act on behalf of users and implements corresponding security/trust models
  - limiting access to trusted friends of its user
  - accessing other agents with its user's persona
- Function 24 by 7 and are never asleep or tired



# Public and Private Agents



- Private agents may be off-line or powered down

- Enabling off-line operation via data synchronization

# Ubiquitous Web Applications

Declarative Authoring with XML

# Benefits from using XML

- Reduced costs for development and maintenance
  - compared to non-declarative techniques
- Improved security, accessibility, usability
- Easier delivery to wide range of devices and platforms
  - through use of a layered architecture
- Facilitate people with different roles to work on different aspects as part of a distributed team
  - allow team members to focus on what they do best

# Building on years of research

- There has been a lot of research into how to build user interfaces over last 15 years
- Model-based
- Multiple layers of abstraction
- Each layer models behavior at a progressively finer level of detail
- Functional transformations between layers
- Use delivery context to select transformation

# Layered UI

with transformations defined between each layer

- 1) Application task and domain models
  - ♦ supported via diagramming languages (UML)
- 2) Abstract User Interface
  - ♦ UI independent, e.g. select 1 from n
- 3) Concrete User Interface
  - ♦ UI specific, e.g. set of radio buttons
- 4) Realization on specific device context
  - ♦ generated via a compilation step
  - ♦ for delivery to HTML, SVG, Flash, Java, .NET

# XML for Concrete UI

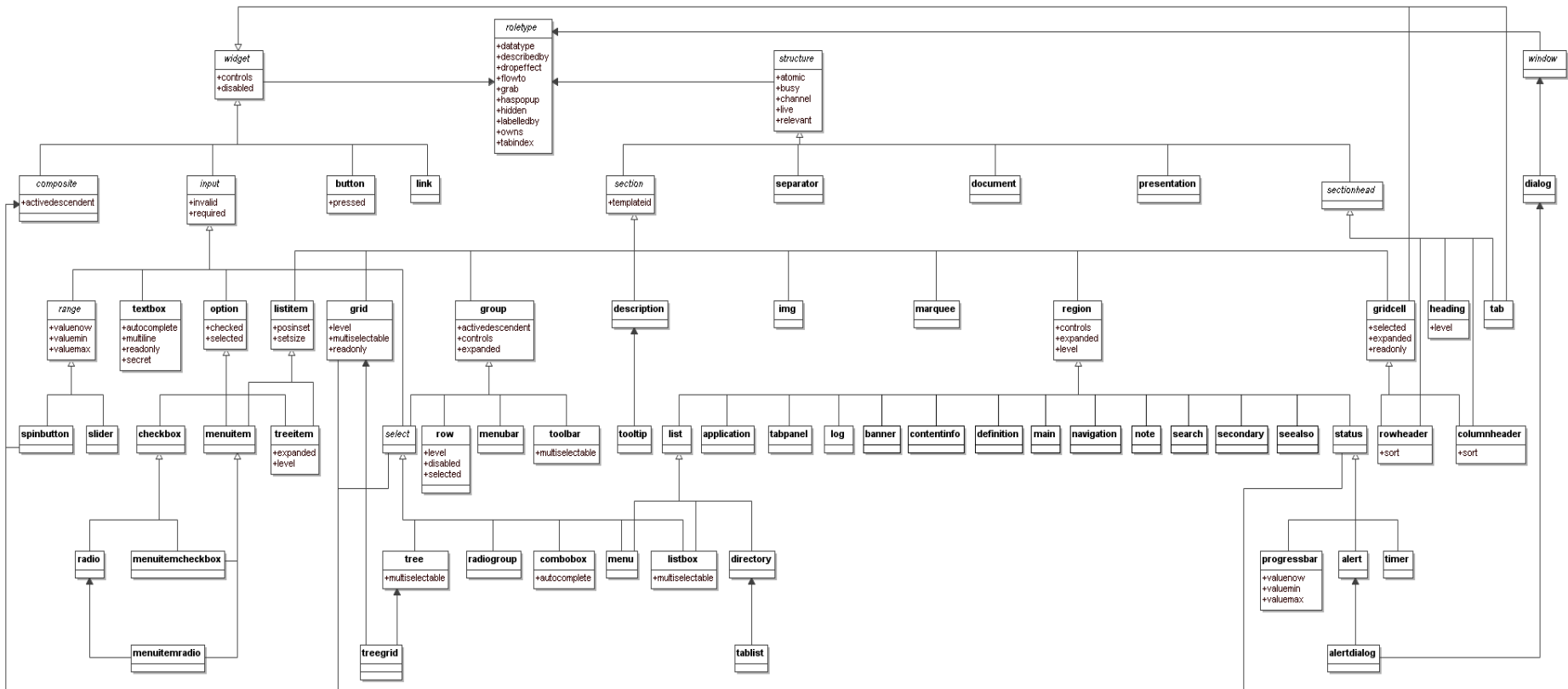
- Use XML for defining UI layout and controls
  - vertical/horizontal/grid layout managers
  - full set of controls e.g. buttons, menus, text input, ...
  - associated concrete UI events
- Themes define details of appearance and behavior on target platforms
- Compile into final UI
  - HTML+JavaScript+CSS
  - Java for JVM (JAR)
  - ActionScript for Flash Player (SWF)

# XML for UI

- Many examples of proprietary UI markup languages, e.g.
  - Microsoft (XAML)
  - Adobe (MXML)
  - Lazlo (OpenLazlo)
  - Nexaweb (XAL)
  - Mozilla (XUL)
- Time for W3C to define an open standard
  - For authoring tools rather than run-time
  - Alignment with accessibility APIs

# WAI-ARIA

- Ontology of UI controls, properties and states
  - Used to enable assistive technology





# 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 people interested in working on
  - device abstraction layer for web applications
  - 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](mailto:dsr@w3.org)>
  - team contact: Stéphane Boyera <[boyera@w3.org](mailto:boyera@w3.org)>

# Ubiquitous Web Applications

Questions?