The Web and Mobile Code

- originally, the Web delivered documents
- now becoming a platform for programs
  - universal GUI interface
- today's agenda
  - background about the Web
  - Java applets
  - ActiveX
  - the future of mobile code

Documents and URLs

- Web stores "documents" (files)
- URL specifies
  - which protocol to use when talking to server
    - » usually HTTP
  - DNS or IP address of server
  - port number of server
  - document pathname on server

HTTP Protocol

- simple protocol for fetching documents
- several client request types
  - GET fetches a file
  - PUT uploads a file to server
  - POST sends a request with arguments
- also supports
  - redirect
  - keep-alive
  - proxies

Proxies

- machine used as intermediary for all requests from a client
- reasons
  - caching
  - security
  - anonymity
  - rewriting or filtering of content
- client speaks proxy-HTTP to proxy
- proxy speaks HTTP to server

Browsers and Document Types

- Web documents are marked with MIME "content types"
  - sometimes inferred from filename extension
- browser has a recipe for displaying each content type
  - some handled by browser itself
  - some handled by browser "plug-in"
  - some handled by external application

Forms

- supported by HTML 2.0
- commands to put various input elements on page
  - checkbox
  - typein field
  - menu
- submission triggered by an input
  - typically use "submit" button
- results sent to server in an HTTP request

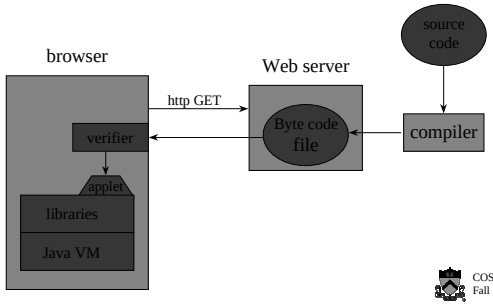
Script Tag

- specifies a simple in-line program to run
- various scripting languages exist
  - JavaScript (=JScript) the most popular
  - VBScript
- scripts can
  - modify properties of browser or page
  - generate HTML dynamically
  - open new windows or browsers

Applet Tag

- <applet height=400 width=600 code=…>
- loads embedded program and gives it a rectangle to run in
- can have multiple applets per page
  - inter-applet communication
- other options
  - archive file
  - serialized applet

Loading Applets



Java Applets

- main class extends java.applet.Applet
  - is a GUI component
  - gets hide/expose/mouse/keyboard events
- other classes loaded on demand, from same Web server
- can call into JavaScript, and vice versa

Applets and Name Spaces

- each applet gets its own name space for classes
  - "built-in" classes shared by all applets
- implemented by built-in AppletClassLoader
  - one AppletClassLoader per applet
  - to resolve a class name
    - » first, see if class defined already
    - » next, loop for built-in class with matching name
    - » finally, try loading from Web server

Stopping Applets

- When do applets die?
- when user clicks to new page, applet is told about it
- applet can choose whether to die, hibernate, or keep running
  - example: ESPN score ticker
  - example: invisible cycle parasite
- can't kill threads forcibly in Java either

Applet Security

- applets are untrusted, so their activities must be controlled
- default "sandbox" policy
  - no file access
  - network access to applet's home machine only
  - no access to environment variables
  - cannot start or monitor processes
  - etc.
- built-in code is unrestricted

Applet Security: Enforcement

- type safety: the foundation
  - no forging of pointers
  - no illegal type casting
  - no illegal access to private/protected variables or methods
- basic elements
  - sound language type system
  - garbage collection
  - dynamic linking

Enforcing Type Safety

- byte-code verification
  - analyzes bytecode
  - infers types of stack and variables at each program point
  - checks for consistency
- class loading / dynamic linking
  - makes sure name->type mappings make sense
- many details to get right in the VM code

Building on the Foundation

- type safety means the only access to sensitive resources is through "official" Java API calls
- official API calls want to deny permission to applets
  - use SecurityManager class to make decisions
- first approach: "who called me?"
- more realistic: "how was I called?"

ActiveX and Plug-Ins

- download executable code
- installed as an addition to the browser
- once installed, can do anything
- can be a platform for other content
- advantages: efficient, flexible
- disadvantage: risky to install

ActiveX/Plug-In Security

- must trust the code provider
- Netscape plug-in approach: user decides based on
  - URL
  - Netscape endorsement
- Microsoft approach: digital signing of ActiveX programs
  - "Authenticode" system

Code Signing

- naïve theory (Authenticode)
  - author digitally signs code
  - consumer's browser verifies signature
  - consumer accepts code if s/he trusts author
- problems
  - signature doesn't mean authorship
  - trust isn't enough

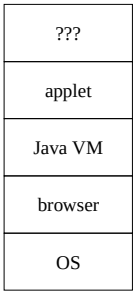
Code Signing, Realistically

- signature implies endorsement
  - "code works as advertised, as far as I can tell"
- endorsement is limited
  - code for use in limited circumstances
  - endorsers accepts limited liability
- accepting signed code is risky
  - must be trustworthy, skilled, and diligent

Code Signing in Java

- loosen restrictive "sandbox" security model for applets signed by trusted parties
- several proposals for how to do this
  - capabilities
  - name space management
  - extended stack inspection
- who makes decisions?
  - user: too complicated and confusing?
  - administrator: one size fits all

Applets as Platforms



- applets often serve as platforms for other content
- applet must define environment and execution rules
- could have even more layers

Network Computers

- small and cheap
  - no disk
  - minimal memory
  - cheap display, or use TV
- all programs delivered as applets
- Java VM and simple OS in ROM
- good: cheap, easy to administrate
- bad: inflexible

The Battle for Desktops

- commercial power goes to the one who controls the platform
  - many ways to leverage platform control
- currently, that's Microsoft
- Java/NC offers an alternative platform
- many commercial and legal battles now over which platform will dominate