Web Architecture

Principles and constraints that characterize Web-based information systems

- **URI**: Uniform Resource Identifier
- **HTTP**: HyperText Transfer Protocol
- Metadata must be recognized and respected
  - Enables making resources comprehensible across administrative domains
  - Difficult to enforce unless the metadata is itself suitably formalized
Uniform Resource Identifier: 1

- URIs are abstract
- What matters is their (purported) uniqueness
- URIs have no proper syntax per se
- Kinds of URIs include
  - URLs, as in browsing: not used in standards any more
  - URNs, which leave the mapping of names to locations up in the air
Uniform Resource Identifier: 2

Good design requirements

- Ensure that the identified resource can be located
- Ensure uniqueness: eliminate the possibility of conflicts through appropriate organizational and technical means
- Prevent ambiguity
- Use an established URI scheme where possible
HTTP: HyperText Transfer Protocol

Intended meanings are quite strict, though not constrained by implementations

- Text-based, stateless
- Key verbs
  - Get
  - Post
  - Put
- Error messages for specific situations, such as resources not available, redirected, permanently moved, and so on

ReST: Representational State Transfer
Representational State Transfer

ReST is an architectural style for networked systems that constrains the connectors

- Models the Web as a network of hyperlinked resources, each identified by a URI
- Models a Web application as a (virtual) state machine
- A client selecting a link effects a state transition, resulting in receiving the next page (next state) of the application
Characteristics of ReST

- Client-Server
- Statelessness: in terms of sessions
  - What is an advantage of statelessness?
  - Where is the session state kept then?
- Focus on resources being manipulated and their representations being transferred
- Uniform Interface: URIs, hypermedia
- Caching: responses can be labeled as cacheable
Basic Interaction Models

Interactions among autonomous and heterogeneous parties

- Adapters: what are exposed by each party to enable interoperation
  - Sensors $\leftrightarrow$ information
  - Effectors $\Rightarrow$ actions
- Invocation-based adapters
- Message-oriented middleware
- Peer-to-peer computing
Invocation-Based Adapters: 1

Distributed objects (EJB, DCOM, CORBA)

- *Synchronous*: blocking method invocation
- *Asynchronous*: nonblocking (one-way) method invocation with callbacks
- *Deferred synchronous*: (in CORBA) sender proceeds independently of the receiver, but only up to a point
Invocation-Based Adapters: 2

Execution is best effort: application must detect any problems

- At most once
- More than once is
  - OK for idempotent operations
  - Not OK otherwise: application must check
DoDAF

Department of Defense Architecture Framework

- A standardized way to organize an enterprise architecture
- Lists 26 views organized into four categories
- Roughly, a software methodology
  - How to capture requirements: user activities
  - How to develop solutions: meet performance criteria
  - How to consider technical standards
- Best for large systems with lifetimes of decades