Derivation of Representational State Transfer (REST)

A hybrid network-based architecture style

Constraints (sub - style)	Benefits	Problems
Client and server	Separation of concern	
Stateless	Visibility, reliability, scalability	Decrease network performance by increasing the repetitive data
Cache	Reducing the average latency of a series of interactions.	May decrease reliability if stale data is used.
The early web architecture (before 1994): client-cache-stateless-server - mainly for the exchange of static documents		
Uniform Interface: - Identification of resources - Manipulation of resources through representations - Self-descriptive messages - Hypermedia as the engine of application state	Generality. Implementations are decoupled from services they provide, which encourages independent evolvability.	Degrades efficiency.
Layered system	Encapsulate legacy services	Overhead and latency
The modern (then) web architecture: Uniform-Layered-Client-Cache-Stateless-Server - supporting dynamically generated content and intermediary components (e.g. proxies)		
Code on demand (optional)		
REST		

Principles of REST

1. The key abstraction of information is a resource, named by an URL. Any information that can be named can be a resource.

2. The representation of a resource is a sequence of bytes, plus representation metadata to describe those bytes. The particular form of the representation can be negotiated between REST components.

3. All interactions are context-free: each interaction contains all of the information necessary to understand the request, independent of any requests that may have preceded it.

4. Components perform only a small set of well-defined methods on a resource producing a representation to capture the current or intended state of that resource and transfer that representation between components. These methods are global to the specific architectural instantiation of REST; for instance, all resources exposed via HTTP are expected to support each operation identically.

5. Idempotent operations and representation metadata are encouraged in support of caching and representation reuse.

6. The presence of intermediaries is promoted. Filtering or redirection intermediaries may also use both the metadata and the representations within requests or responses to augment, restrict, or modify requests and responses in a manner that is transparent to both the user agent and the origin server.

Terms

Distributed Hypermedia System (E.g., Web)

Network-based architecture styles

Resource
- A resource is a computational mapping to a set of entities, not the entity that corresponds to the mapping at any particular point in time.
- Why so abstract: to be general enough to encompass various information.

Representations
- REST components perform actions on a resource by using a representation to capture the current or intended state of the source and transferring that representation between components.
- A representation consists of data, metadata describing the data.
- Web pages we see every day online are representations, not resources!