Material and some slide content from:

- Software Architecture: Foundations, Theory, and Practice
- Krzysztof Czarnecki

#### Cloud Computing, and REST-based Architectures Reid Holmes



**REID HOLMES - SE2:** SOFTWARE DESIGN & ARCHITECTURE

## Cloud precursors

- Grid Computing:
  - Combination of computing resources from multiple administrative domains applied to common tasks.
    - Usually used to create 'super computers' that can work on specific parallel computation tasks.
- Utility Computing:
  - Combining computation, storage, and services metered like utilities.



# Cloud Computing

 "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models." [NIST]



#### NIST Essential Characteristics

- On-demand self-service:
  - Consumers can provision computing capabilities without human interaction.
- Broad network access:
  - Capabilities are available over the network through standard mechanisms.
- Resource pooling:
  - Computing resources are pooled to serve multiple consumers.
  - Location independence. [perfomance/security]



#### NIST Essential Characteristics

- Rapid elasticity
  - Resources can be easily added and removed.
- Measured service [services and/or resources]
  - Metering of storage, processing, bandwidth, etc.



## Benefits

- Agility [quickly respond to changes]
- Scalability [resources can be added, peak load]
- Cost [resources can be released; multi-tenancy (amortization)]
- Reliability [different sites, experts in control]
- Security [works both ways]

# Cloud Layers

- SaaS (e.g., Google Docs) [multi-tenancy, single release for all users]
  - Vendor-controlled remote applications.
  - Concerns: control, performance, security, privacy.
- PaaS (e.g., AppEngine)
  - Vendor-controlled environment.
  - Concerns: as for SaaS w/ limited technology choices.
- IaaS (e.g., Amazon EC2)
  - Vendor-provided resources; consumer provisions VM.
  - Concerns: more expertise needed to leverage flexibility.



## Cloud Spectrum

2

less flexible more constrained less effort

#### more flexible less constrained more effort





#### Layers of Control



REID HOLMES - SE2: SOFTWARE DESIGN & ARCHITECTURE

# Cloud Security NFPs

- Users want assurances of:
  - Confidentiality [keep unauthorized users out]
  - Integrity [data has not altered]
  - Authenticity [data provenance]
  - Anonymity [users are unidentifiable]
  - Privacy [user data is properly controlled]
- Data remanence is problematic:
  - How can you purge data from the cloud?



#### REST

- Representational state transfer (REST)
- Key constraints:
  - Client/server
  - Servers to not maintain session state
  - Clients must not depend on direct server access
  - Clients communicate using a uniform interface
    - e.g., URIs and self-descriptive payloads



# **REST** Operations

- ► Four main operations: GET, POST, PUT, DELETE
- Operation can change functionality:
  - ► GET /resources/ —> list resources
  - PUT /resources/ —> replace resources
  - POST /resources/ —> append to resources
  - DELETE /resources/ —> delete resources
- URIs are often versioned:
  - /api/v2.0/list/
  - /api/v3.0/list/
- REST endpoints enable direct testing:
  - e.g., curl --include https://api.github.com/users/rtholmes