Cloud Platform Support for API Governance
o “Service-ize” digital assets and IP
o Accessible everywhere, all the time (BYOD)

o **Platforms-as-a service (Paas)** simplify this process
  o Facilitates deployment for software/data as-a-service
    o Automates configuration, deployment, container management, monitoring, …
- Code and data of value to the organization
- Application Programming Interface (API)
  - Common entry point for access & control
    - By client apps and browsers
  - Functional boundary
- Web service software environment (SW Env)
  - Storage technologies: SQL, NoSQL, bucket/object stores, HDFS
  - Computation technologies: Hadoop, tasks, event handlers
  - Load balancers, app servers

Storage technologies: SQL, NoSQL, bucket/object stores, HDFS
Computation technologies: Hadoop, tasks, event handlers
Load balancers, app servers
- Code and data of value to the organization
- Application Programming Interface (API)
  - Common entry point for access & control
    - By client apps and browsers
  - Functional boundary
- Web service software environment (SW Env)
  - Storage and computational technologies
- API lifecycle
  - Evolves with that of digital assets (DAs)
  - Software environment can/does evolve separately
    - Can change without impacting API clients
- Popular development model
  - Service-ize digital assets, exposing them via APIs
  - Reuse extant APIs to construct new APIs, mashups, applications
- For public consumption
  - Pay-per-use and free
  - Google, Facebook, Twitter, Yahoo!, …
- Popular development model
  - Service-ize digital assets, exposing them via APIs
  - Reuse extant APIs to construct new APIs, mashups, applications
- For public consumption
  - Pay-per-use and free

- **Within organizations**
  - Leverage development across organization through **reuse**
    - Internal and external development
  - Agile processes and DevOps change SW environment frequently

- **API becomes point of strategic business decisions**
  - Common entry point into org (security, access control, activity)
  - Focuses development, DevOps, and IT
    - Can influence business value
o HW/SW infrastructure now commoditized by cloud

o **Experiencing a shift** toward management of
  o Software environment (directly controllable by DevOps)
  o APIs
    o Control, maintain, facilitate reuse, and secure
    o Provision resources and software environment

o Requires new tools and a system-wide framework
  o For **API Governance**: combined policy, implementation, and deployment control of APIs for IT-managed services and DAs
    o Unify and automate API management processes
    o Facilitate efficiency **and scale**
      o In number of APIs, API clients (service users), developers
    o That accounts for input from business concerns
- PaaS is the perfect infrastructure for API Governance
  - Deployment automation, elasticity, fault tolerance, high availability, logging and monitoring, on-premise and/or public

- Requires additional support for
  - API cataloging, search, and registration
    - This already exists in the numerous API management platforms!

- **Unified and automated policy support**
  - Specification (a language)
  - Verification, analysis, and feedback (developer tools)
  - Enforcement
    - Deployment time
    - Runtime
API Governance for IT Managed Infrastructure

Service consumers and clients

• Static analysis
  • Policy verification
  • Automated testing
  • Autogen of enforcement logic

• Dependency checks
  • APIs
  • SW infrastructure
  • Provenance tracking

• Runtime policy enforcement
  • Access and rate control
  • AB testing
  • Auditing & feedback gathering
  • SLA & QOS checks

EAGER -- Enforced API Governance for REST

SW environment

Developers

Deployment Enforcement

Runtime Enforcement

Dev Tools
- PaaS platform that decouples innovation from common services
  - Automatically manages and scales apps + service ecosystem
  - Access scalable services via well-defined de facto standard APIs
- PaaS platform that decouples innovation from common services
  - Automatically manages and scales apps + service ecosystem
  - Access scalable services via well-defined de facto standard APIs
    - Starting point: Google App Engine
NO CODE REWRITE

On-premise
Behind your firewall
Everywhere

AppScale

Your App Here

Google app engine

docker

Eucalyptus

Amazon web services

VMware

KVM

UCSB

Windows Azure
- Enforced API Governance for REST
  - Policy language
    - Restricted subset of Python
    - Policies: per-operation, per-API, system-wide
  - Developer Tools
    - API analysis
      - Static policy verification
      - Automatic generation of
        - Functional tests from policies
        - Policies from unit tests
        - Deployment enforcement checks
        - Runtime deployment checks
  - Deployment enforcement
  - Runtime enforcement
API Similarity Tool
- Evaluate the “porting effort” associated with changing an application
  - That is using one API, to use a similar API
- Helps developers and IT managers reason about
  - How hard it will be to change to use a similar API (reuse code)
  - How similar two APIs are (for policy enforcement @ code reuse)
  - How APIs evolve over time (and how to enforce change control)
- Describe API behavioral and functional semantics
  - Using the EAGER language: as axiomatic semantics
  - Translate to ASTs (per operation)
  - Employ DICE coefficient (Hoare’s Rule of Consequence)
    - To compute an AST similarity (porting effort) score
Empirical Valuation: Porting Effort
- API Governance is increasingly important for IT
  - Vast proliferation of API development/deployment
  - Lacking management, control, and automation
  - API Governance is unified and automated API policy specification, analysis, auditing, and control
- PaaS is the ideal foundation for providing API Governance solution
  - Elasticity, fault tolerance, scale, distribution, portability
- EAGER extends PaaS (AppScale in particular) with
  - Policy specification, verification, and enforcement
  - Developer tools (analysis, feedback, autogeneration of tests and enforcement checks)
  - Automatic deployment and runtime enforcement
Recent Student Researchers and Visitors!

- **Current**: Stratos Dimopoulos, Geoffrey Douglas, Adam Ehrlich, Chris Horuk, Hiranya Jayathilaka, Alex Pucher
- **Past**: V. Arora, M. Baranski, C. Bunch, N. Canumalla, J. Chohan, N. Chohan, A. Gupta, S. Hedge, M. Hubert, J. Kupferman, P. Lakhina, Y. Li, Y. Nomura (Fujitsu), K. Prakasam, S. Sundaram

Collaborators

- Linda Petzold (CSE/UCSB), Andreas Hellander (Uppsala U), Rich Wolski (UCSB/Eucalyptus)

Support

- Google, IBM Research, NSF, NIH

http://www.cs.ucsb.edu/~ckrintz  
ckrintz@cs.ucsb.edu
http://www.appscale.com (AppScale Systems)