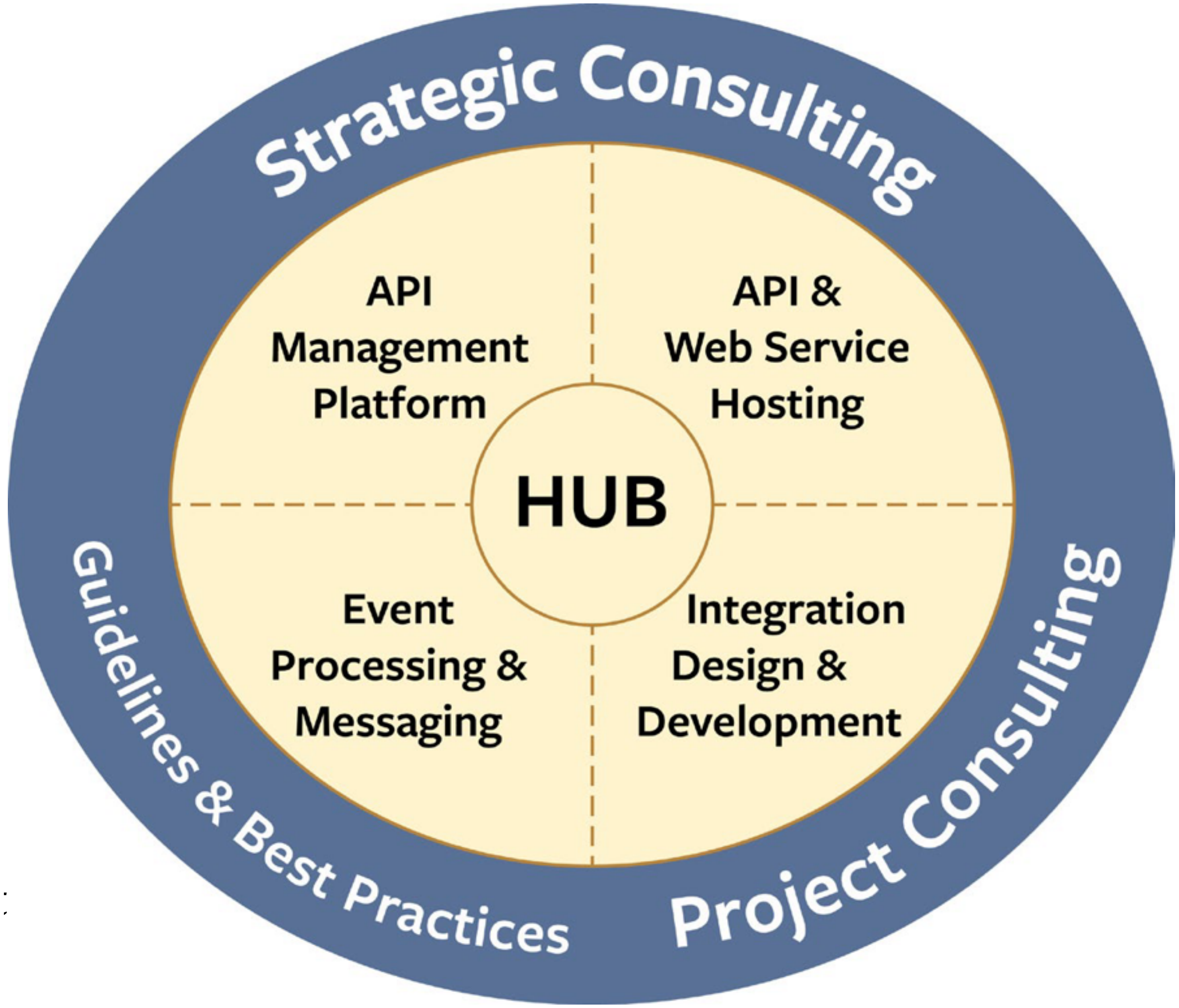
Security in the Cloud @ UC Berkeley

CSG 16.01 Cloud Workshop
(Reprise & Update of CSG 15.01 Short Discussion 2)

William Allison   wallison@berkeley.edu

Berkeley
UNIVERSITY OF CALIFORNIA

The Campus API Service …In the Cloud



Strategic Consulting
API Management Platform
API & Web Service Hosting
HUB
Event Processing & Messaging
Integration Design & Development
Guidelines & Best Practices
Project Consulting

Implementing APIs for PeopleSoft Campus Solutions

Berkeley
UNIVERSITY OF CALIFORNIA

Campus partners

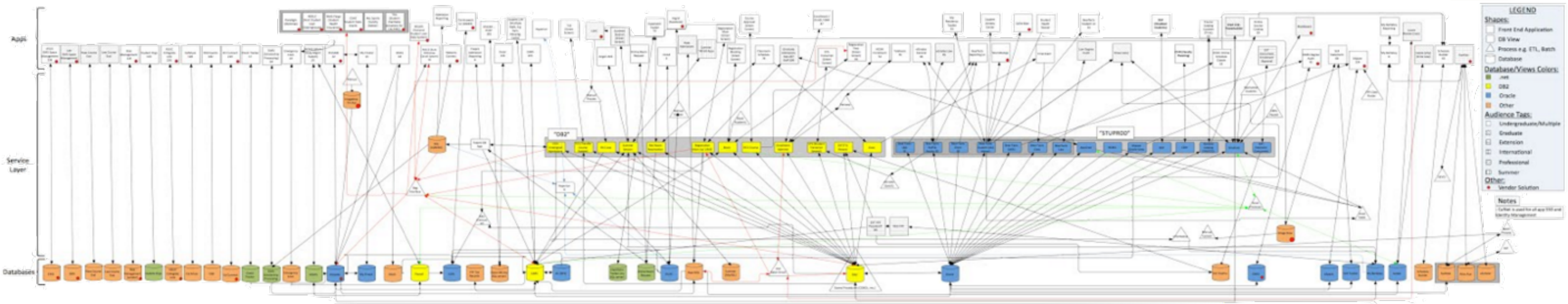
Berkeley | Student Information Systems Project

CALCENTRAL

BEAR FACTS

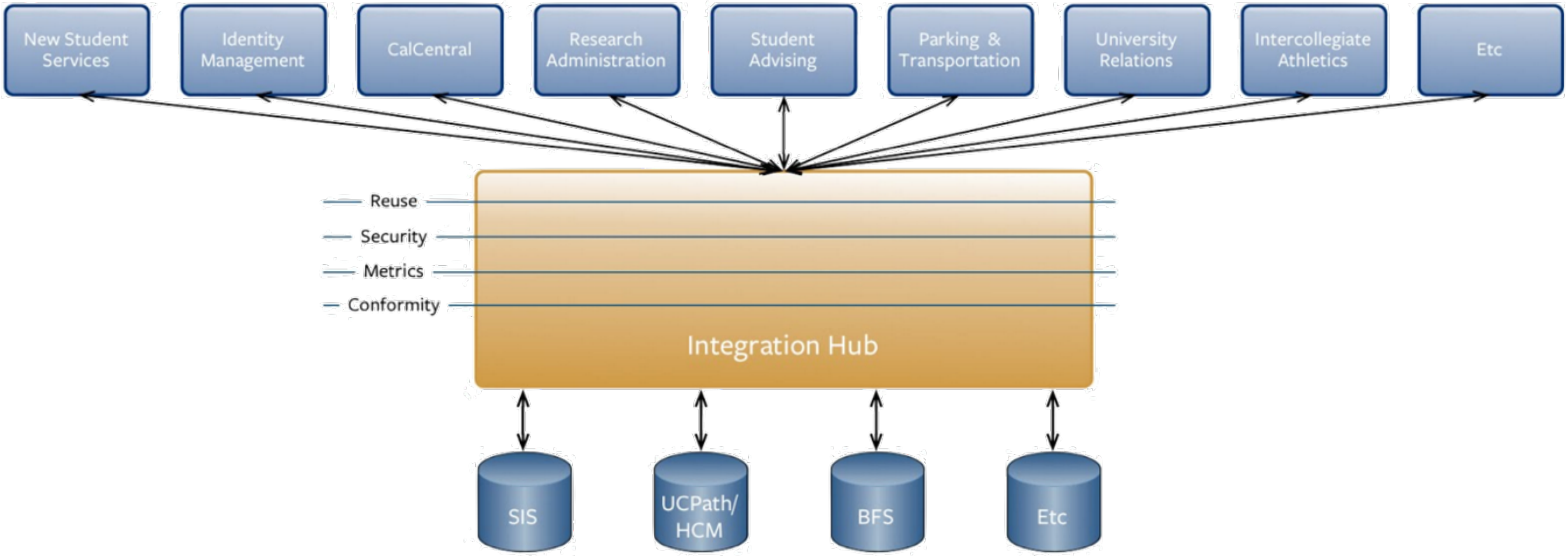
CalNet
Identity and Access Management

THE OLD WAY (EXPENSIVE, FRAGILE): Pre-Enterprise Integration Services SIS Logical Architecture Model (LAMI):



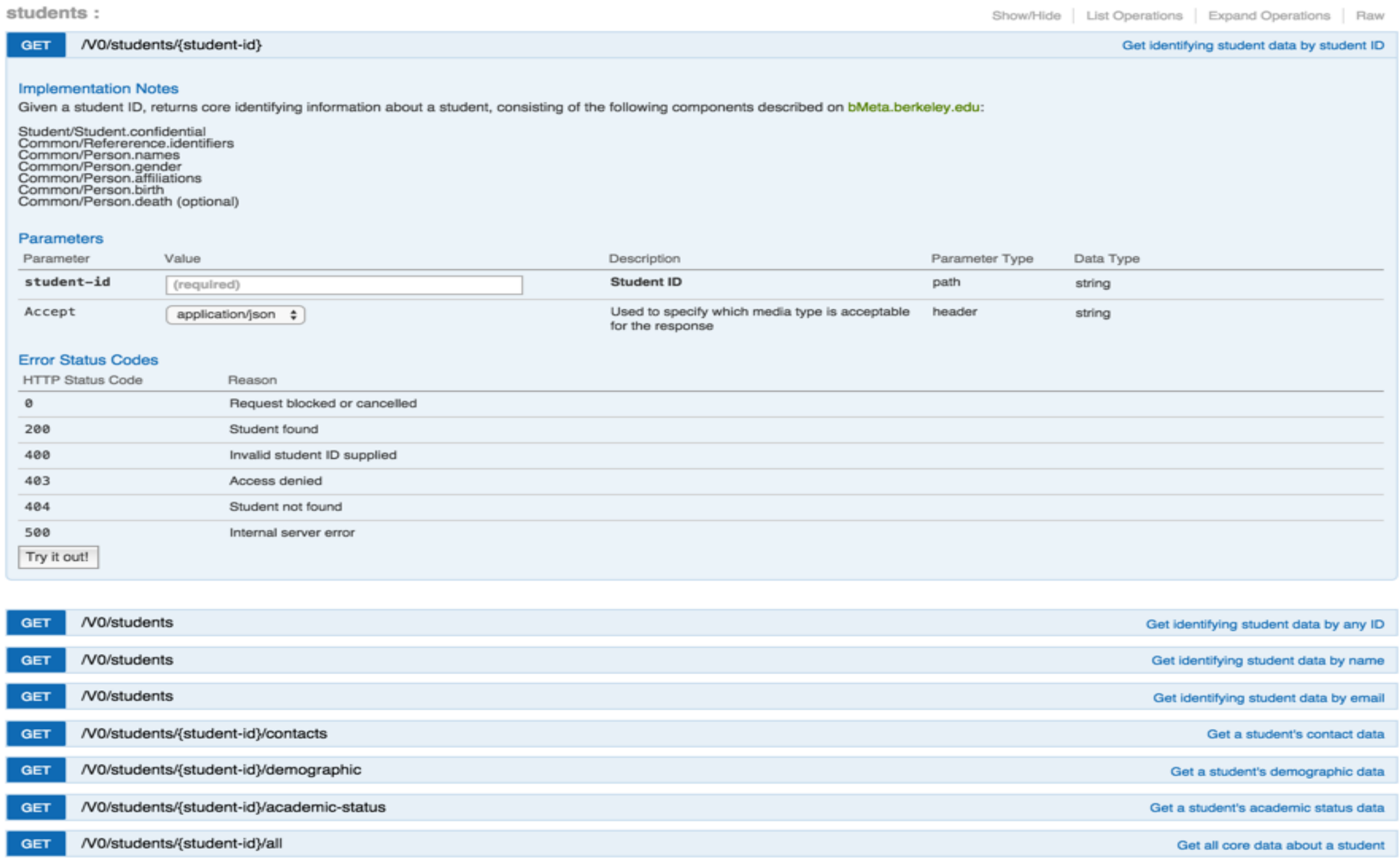
THE NEW WAY (AGILE, COST EFFECTIVE): IST Enterprise Integration Services Campus Integration Model:

Featuring better code reuse, security and privacy, usage metrics, and data conformity transparently as part of every access:



Berkeley
UNIVERSITY OF CALIFORNIA

API Central



students :

GET /V0/students/{student-id} — Get identifying student data by student ID

Implementation Notes

Given a student ID, returns core identifying information about a student, consisting of the following components described on bMeta.berkeley.edu:

Student/Student.confidential
Common/Refererence.identifiers
Common/Person.names
Common/Person.gender
Common/Person.affiliations
Common/Person.birth
Common/Person.death (optional)

Parameters

Parameter	Value	Description	Parameter Type	Data Type
`student-id`	(required)	**Student ID**	path	string
`Accept`	application/json	Used to specify which media type is acceptable for the response	header	string

Error Status Codes

HTTP Status Code	Reason
0	Request blocked or cancelled
200	Student found
400	Invalid student ID supplied
403	Access denied
404	Student not found
500	Internal server error

GET /V0/students — Get identifying student data by any ID

GET /V0/students — Get identifying student data by name

GET /V0/students — Get identifying student data by email

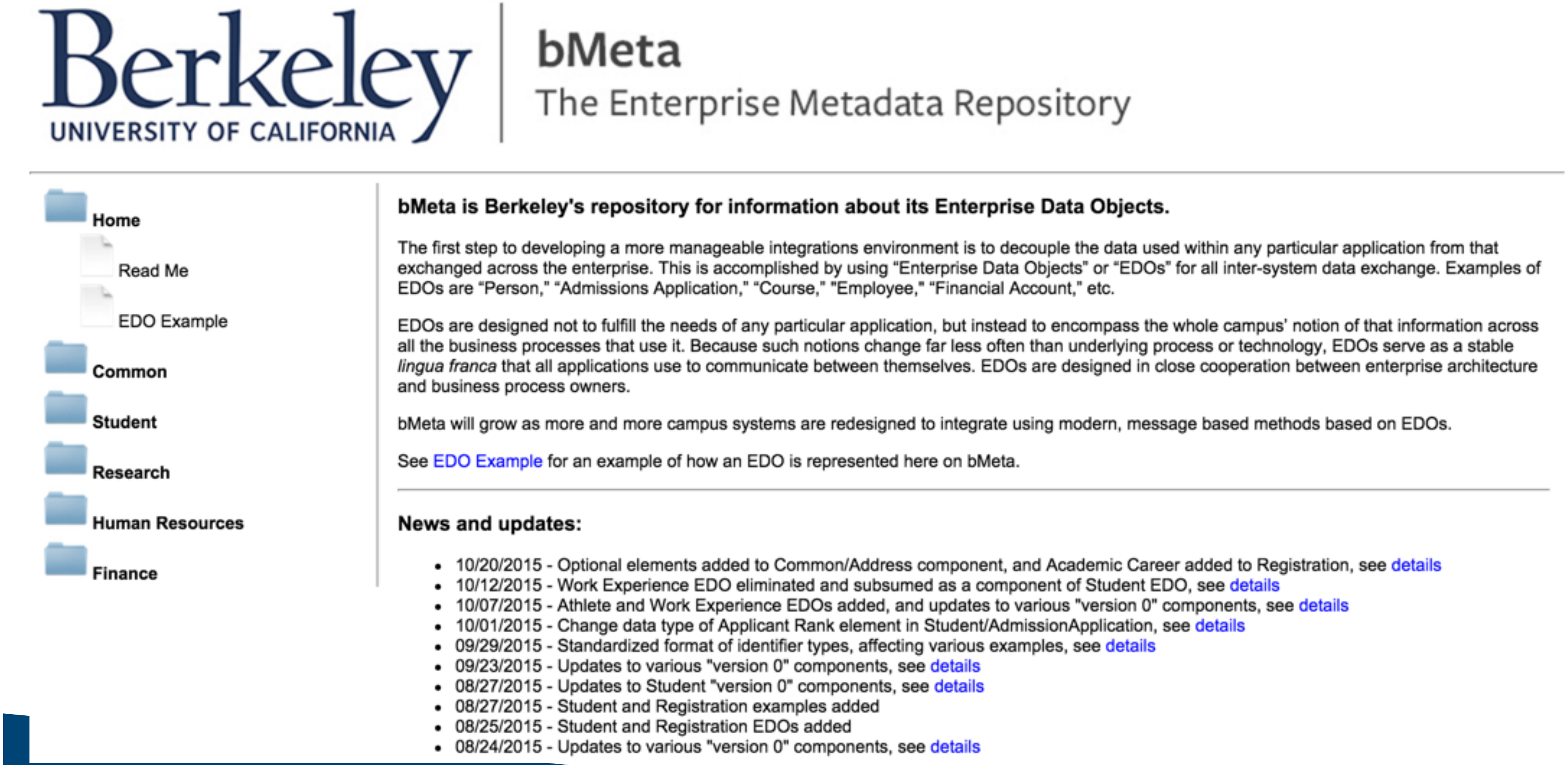
GET /V0/students/{student-id}/contacts — Get a student's contact data

GET /V0/students/{student-id}/demographic — Get a student's demographic data

GET /V0/students/{student-id}/academic-status — Get a student's academic status data

GET /V0/students/{student-id}/all — Get all core data about a student

Metadata repository, hosted on AWS



bMeta

The Enterprise Metadata Repository

bMeta is Berkeley's repository for information about its Enterprise Data Objects.

The first step to developing a more manageable integrations environment is to decouple the data used within any particular application from that exchanged across the enterprise. This is accomplished by using "Enterprise Data Objects" or "EDOs" for all inter-system data exchange. Examples of EDOs are "Person," "Admissions Application," "Course," "Employee," "Financial Account," etc.

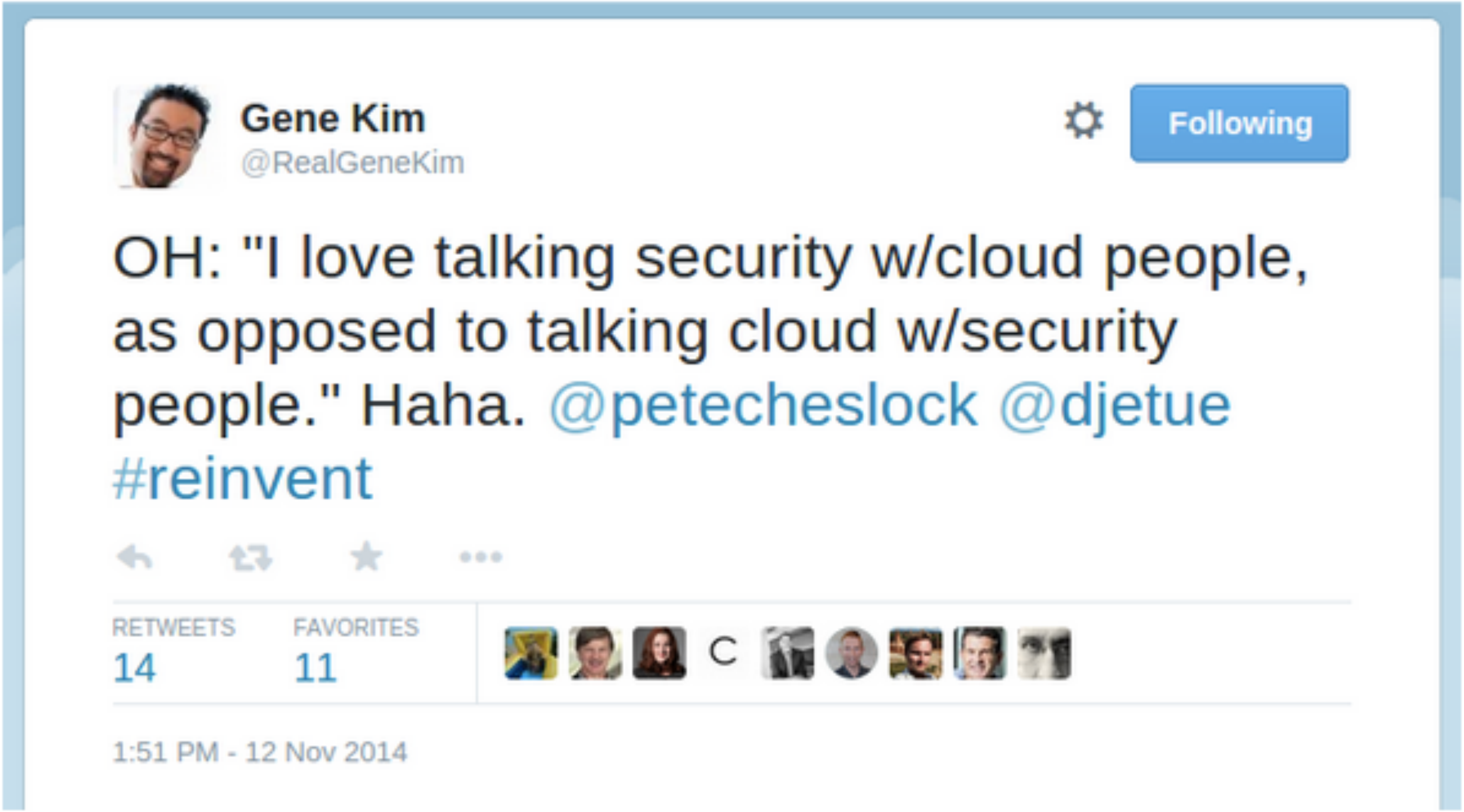
EDOs are designed not to fulfill the needs of any particular application, but instead to encompass the whole campus' notion of that information across all the business processes that use it. Because such notions change far less often than underlying process or technology, EDOs serve as a stable *lingua franca* that all applications use to communicate between themselves. EDOs are designed in close cooperation between enterprise architecture and business process owners.

bMeta will grow as more and more campus systems are redesigned to integrate using modern, message based methods based on EDOs.

See EDO Example for an example of how an EDO is represented here on bMeta.

News and updates:

- 10/20/2015 - Optional elements added to Common/Address component, and Academic Career added to Registration, see details
- 10/12/2015 - Work Experience EDO eliminated and subsumed as a component of Student EDO, see details
- 10/07/2015 - Athlete and Work Experience EDOs added, and updates to various "version 0" components, see details
- 10/01/2015 - Change data type of Applicant Rank element in Student/AdmissionApplication, see details
- 09/29/2015 - Standardized format of identifier types, affecting various examples, see details
- 09/23/2015 - Updates to various "version 0" components, see details
- 08/27/2015 - Updates to Student "version 0" components, see details
- 08/27/2015 - Student and Registration examples added
- 08/25/2015 - Student and Registration EDOs added
- 08/24/2015 - Updates to various "version 0" components, see details



Gene Kim
@RealGeneKim

OH: "I love talking security w/cloud people, as opposed to talking cloud w/security people." Haha. @petecheslock @djetue #reinvent

RETWEETS 14

FAVORITES 11

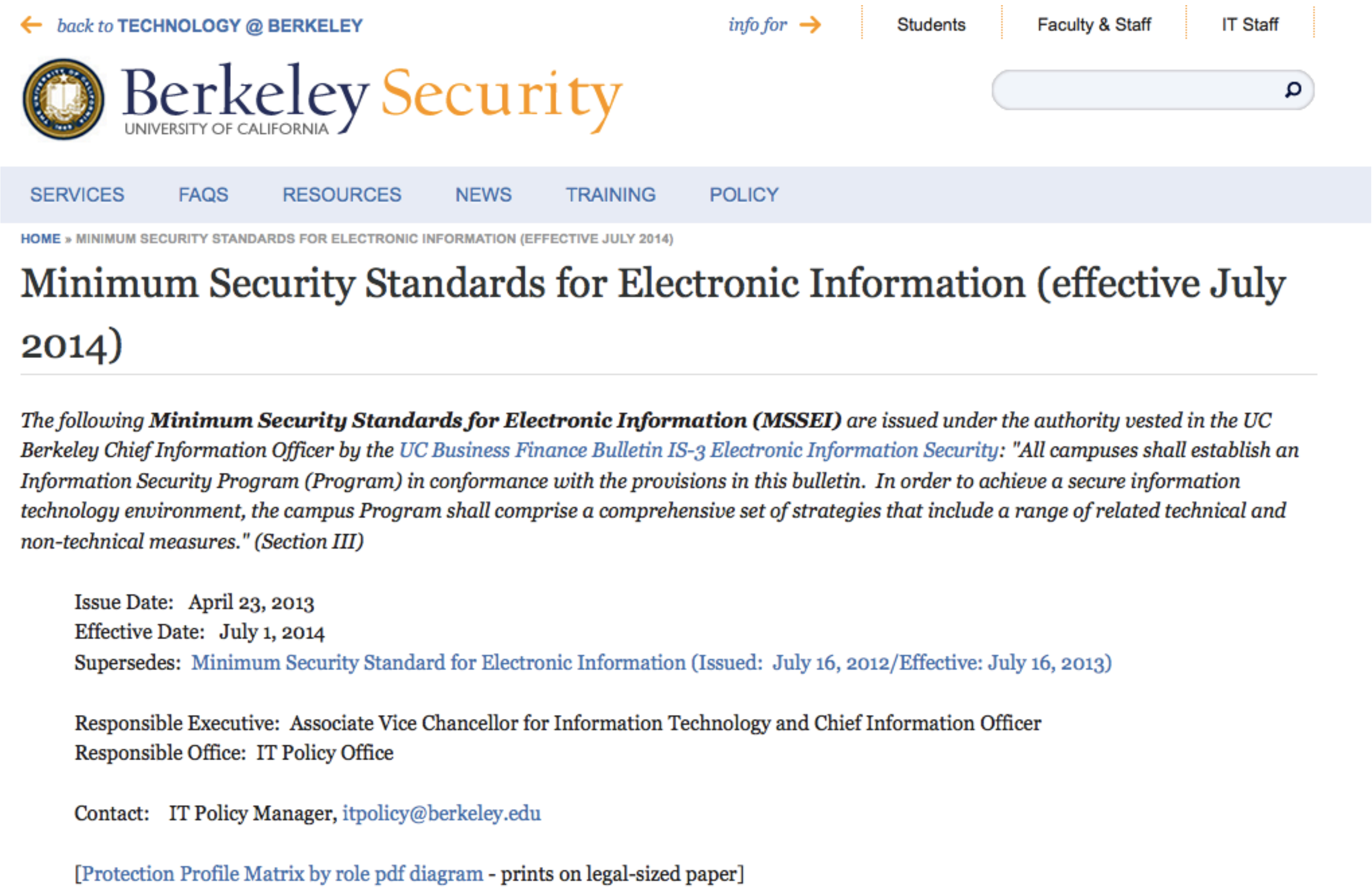
1:51 PM - 12 Nov 2014

The Security Review Process:

https://security.berkeley.edu/data-classification

- Classify Data
- Architect system for actual security
- Review requirements for DPL
- Amend architecture
- Submit MSSEI Self-Assessment
- Iterate over concerns raised by ISP

https://security.berkeley.edu/mssei



Minimum Security Standards for Electronic Information (effective July 2014)

*The following **Minimum Security Standards for Electronic Information (MSSEI)** are issued under the authority vested in the UC Berkeley Chief Information Officer by the UC Business Finance Bulletin IS-3 Electronic Information Security: "All campuses shall establish an Information Security Program (Program) in conformance with the provisions in this bulletin. In order to achieve a secure information technology environment, the campus Program shall comprise a comprehensive set of strategies that include a range of related technical and non-technical measures." (Section III)*

Issue Date: April 23, 2013
Effective Date: July 1, 2014
Supersedes: Minimum Security Standard for Electronic Information (Issued: July 16, 2012/Effective: July 16, 2013)

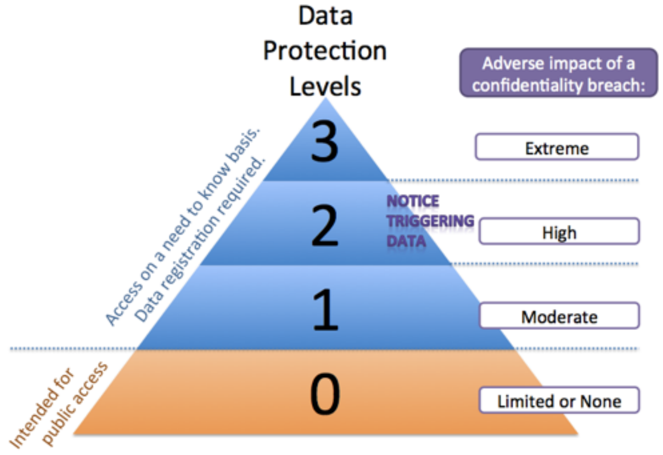
Responsible Executive: Associate Vice Chancellor for Information Technology and Chief Information Officer
Responsible Office: IT Policy Office

Contact: IT Policy Manager, itpolicy@berkeley.edu

[Protection Profile Matrix by role pdf diagram - prints on legal-sized paper]

Data Classification Standard

http://security.berkeley.edu/data-classification


Data Protection Levels
Adverse impact of a confidentiality breach:
3
Extreme
2
NOTICE TRIGGERING DATA
High
1
Moderate
0
Limited or None
Access on a need to know basis.
Data registration required.
Intended for public access

Data Protection Level	Adverse impact*	Sample Data Types (not an exhaustive list)
Level 3	Extreme	Data that creates extensive "shared-fate" risk between multiple sensitive systems, e.g., enterprise credential stores, backup data systems, and central system management consoles.
Level 2	High	Data elements with a statutory requirement for notification to affected parties in case of a confidentiality breach: • **Social security number** • **Driver's license number, California identification number** • **Financial account numbers, credit or debit card numbers; financial account security codes, access codes, or passwords** • **Personal medical information** • **Personal health insurance information**
Level 1	Moderate	Information intended for **release only on a need-to-know basis**, incl.: Personal information not otherwise classified as Level 0, 2 or 3, and Data protected or restricted by contract, grant, or other agreement terms and conditions, e.g.,: • FERPA student records (including Student ID) • Staff and academic personnel records (including Employee ID) • Licensed software/software license keys • Library paid subscription electronic resources
Level 0	Limited or None	Information **intended for public access**, e.g.,: Public websites, Course listings and pre-requisites, and Public directory data: **Staff:** Name, Date of hire, Current position title, Current salary, Organizational unit assignment, Date of separation, Office address, Office telephone number, Current job description, Full-time or part-time, and Appointment type **Students (unless the student has requested that information about them not be released as public information):** Name, Address, Telephone, Email, Dates of attendance, Number of course units in which enrolled, Class level, Major field of study, Last school attended, Degrees and honors received, Participation in official student activities, Weight/height (intercollegiate athletic team members only)

Public records requests, litigation or other legal obligations may require disclosure of information in any data class.

Self Assessment – Step 1

WHAT IS IT?

http://api-central.berkeley.edu

WHAT DOES IT DO?

Together the Nginx Reverse Proxy Service and the 3Scale vendor product form a platform that enables APIs to be easily discoverable, well-documented, easy to use, secured, monitored, and metered. API consumers can find and explore APIs on the API Central portal, where reverse proxy simplifies and standardizes endpoint URIs. API providers and data stewards can control access to an API using the API Central Portal's credentialing service, and can limit usage and mitigate abuse using its metering service.

Risk Classification

“After consulting with others in Security, we will be classifying the 3Scale system as a PL3. The reason for the elevated classification is because having credentials (even for short time period of time) to multiple PL2 systems will create a “shared fate” and warrants the elevation.”

Step 2 – Target Audience

Describe the users who will use and be affected by the application.

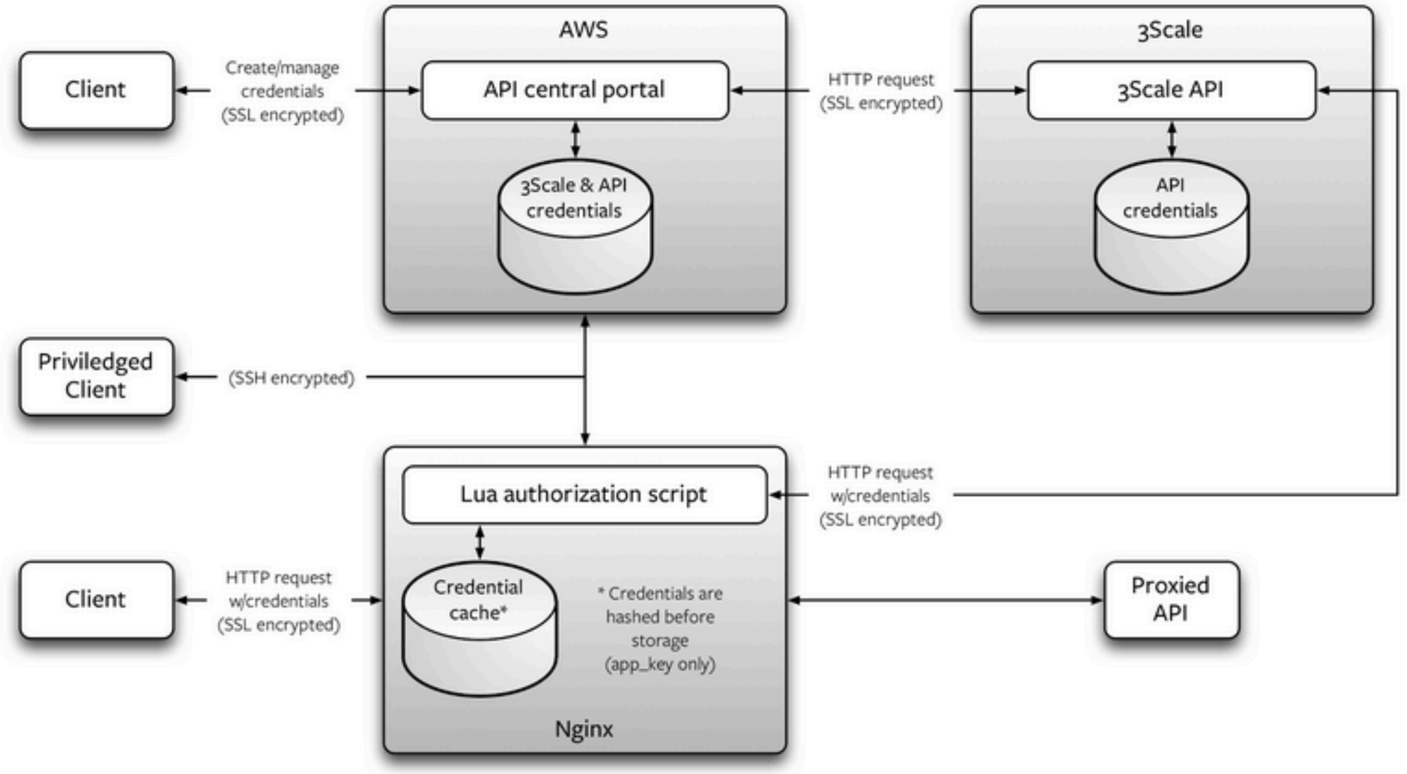
The customers for this API Management and Support Service are system-of-record stewards who provide APIs to access their data and developers who wish to call those APIs.
Currently the APIs are REST based, and are almost entirely read-only (using the http GET method). Requests that update data on the backend sources can be identified by use of the http methods POST, DELETE or PUT. They would however go through the same URL endpoints - this core to the semantics of REST APIs.

We are definitely planning to allow APIs that update state on the backend - what exactly gets updated depends on the the particular API involved.

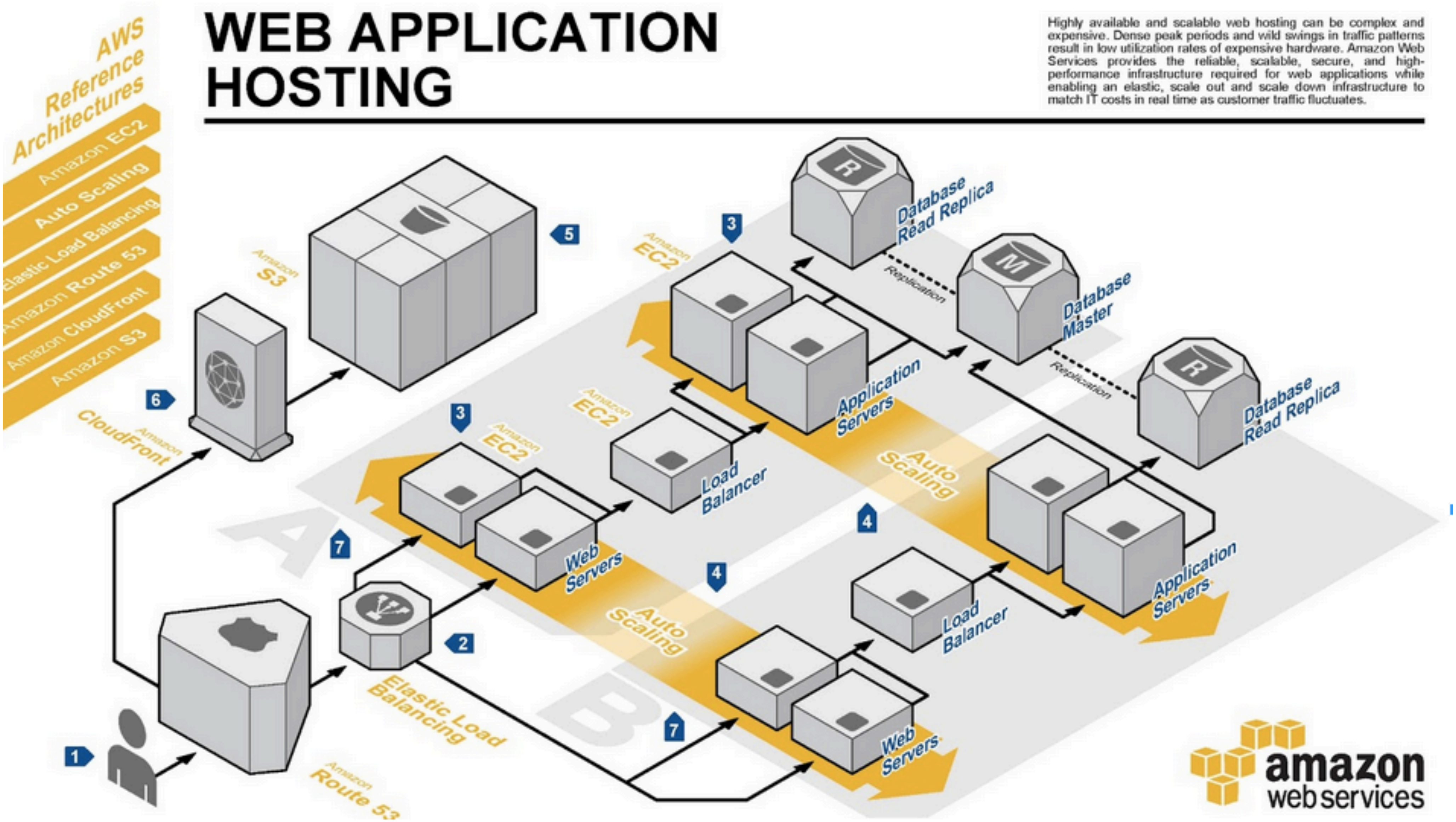
Among the currently deployed APIs, only the Easy Messaging Service allows updating state via the PUT method. Performing a PUT doesn't update any system configuration, but does add an entry into an application message queue.

Step 3 – Architecture Model

Attach a high-level diagram of data flow and data storage, including all the interconnected system names and interfaces.


AWS
API central portal
3Scale & API credentials
3Scale
3Scale API
API credentials
Client
Create/manage credentials (SSL encrypted)
HTTP request (SSL encrypted)
Priviledged Client
(SSH encrypted)
Lua authorization script
Credential cache*
* Credentials are hashed before storage (app_key only)
Nginx
HTTP request w/credentials (SSL encrypted)
Client
HTTP request w/credentials (SSL encrypted)
Proxied API

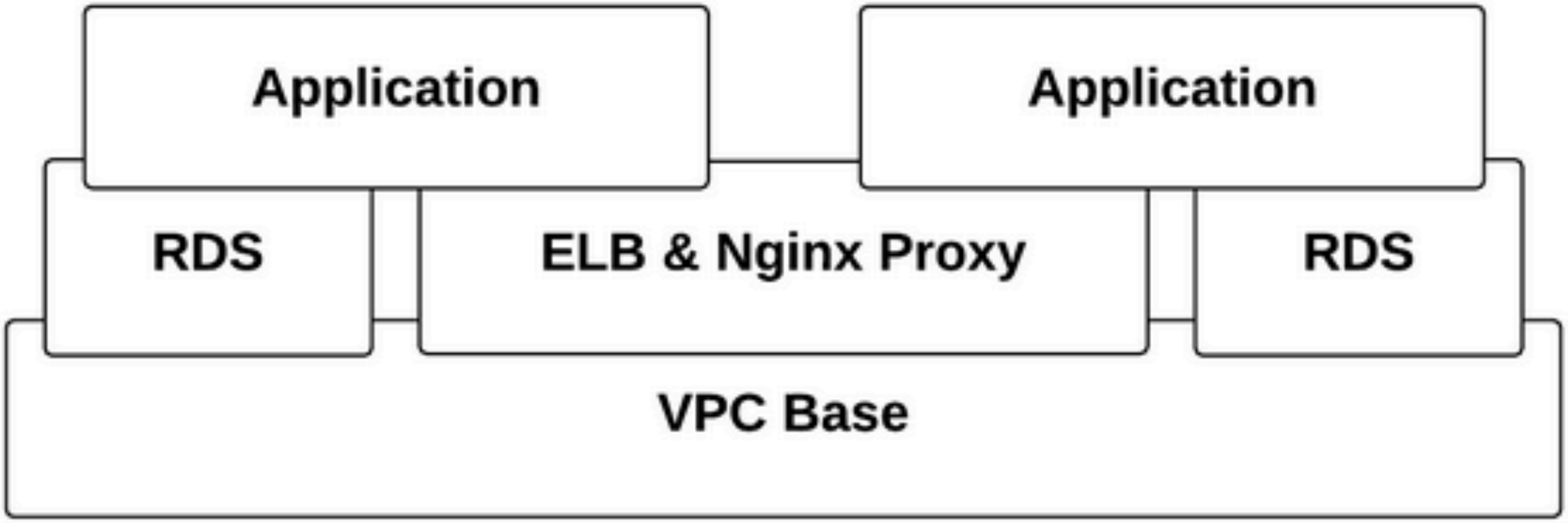
Berkeley
UNIVERSITY OF CALIFORNIA



WEB APPLICATION HOSTING

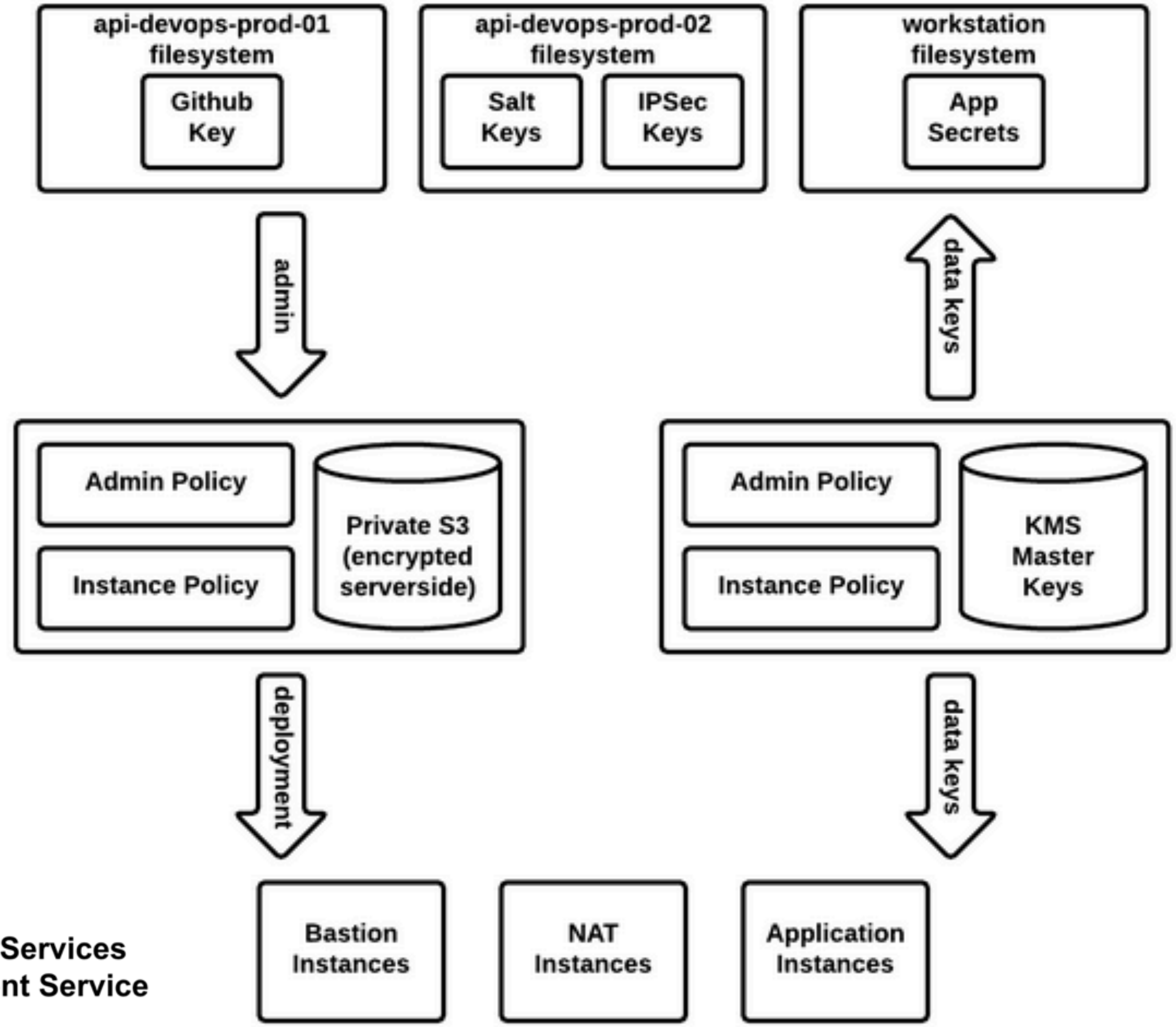
Highly available and scalable web hosting can be complex and expensive. Dense peak periods and wild swings in traffic patterns result in low utilization rates of expensive hardware. Amazon Web Services provides the reliable, scalable, secure, and high-performance infrastructure required for web applications while enabling an elastic, scale out and scale down infrastructure to match IT costs in real time as customer traffic fluctuates.

CloudFormation Stack Layering



CloudFormation

Berkeley
UNIVERSITY OF CALIFORNIA



api-devops-prod-01 filesystem

Github Key

api-devops-prod-02 filesystem

Salt Keys

IPSec Keys

workstation filesystem

App Secrets

admin

data keys

Admin Policy

Instance Policy

Private S3 (encrypted serverside)

Admin Policy

Instance Policy

KMS Master Keys

deployment

data keys

Bastion Instances

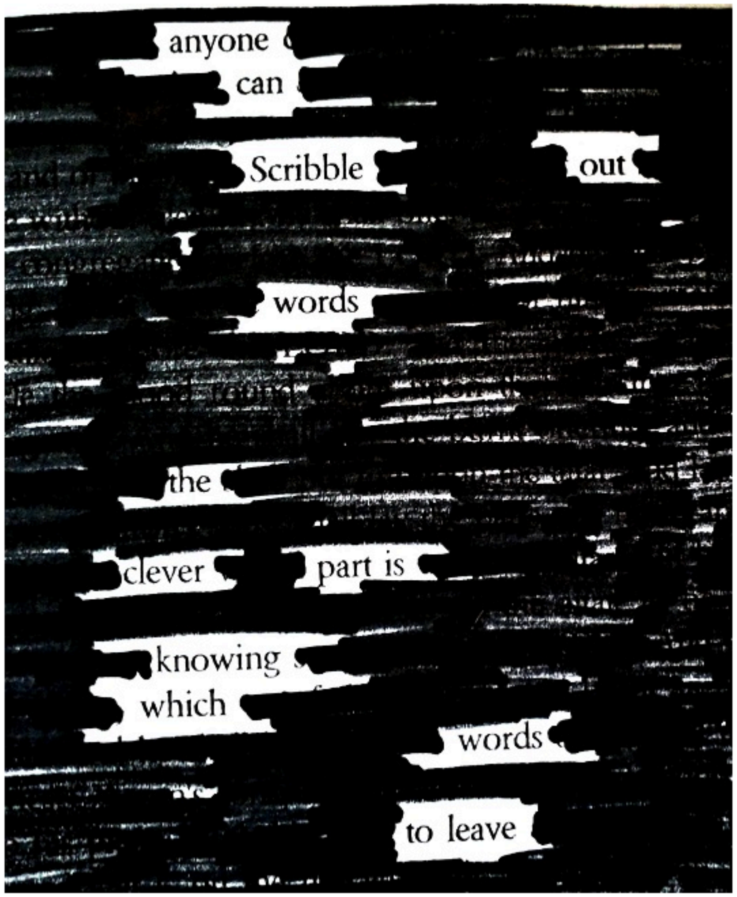
NAT Instances

Application Instances

Amazon Web Services
Key Management Service

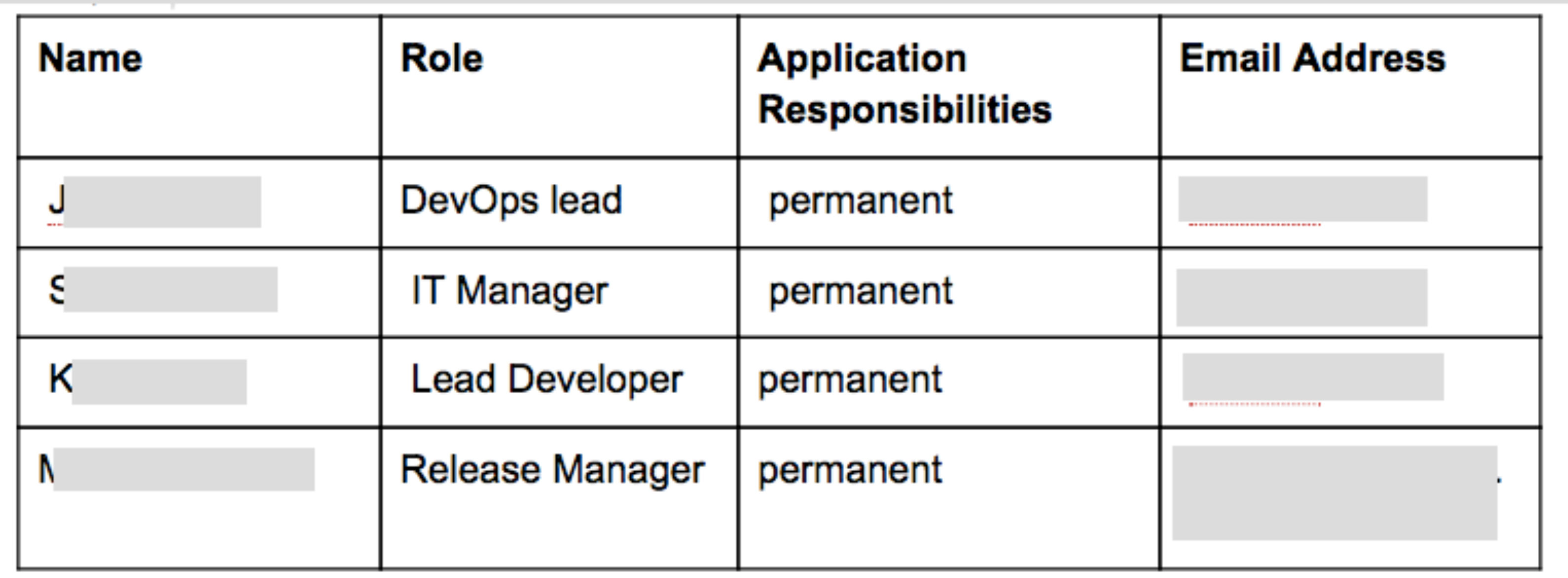
Step 4 – Data Flow Description

Provide description of data movement and data storage depicted in the architecture model. Please include brief description of how each component in the architecture model is being secured.



Step 5 - Support Model

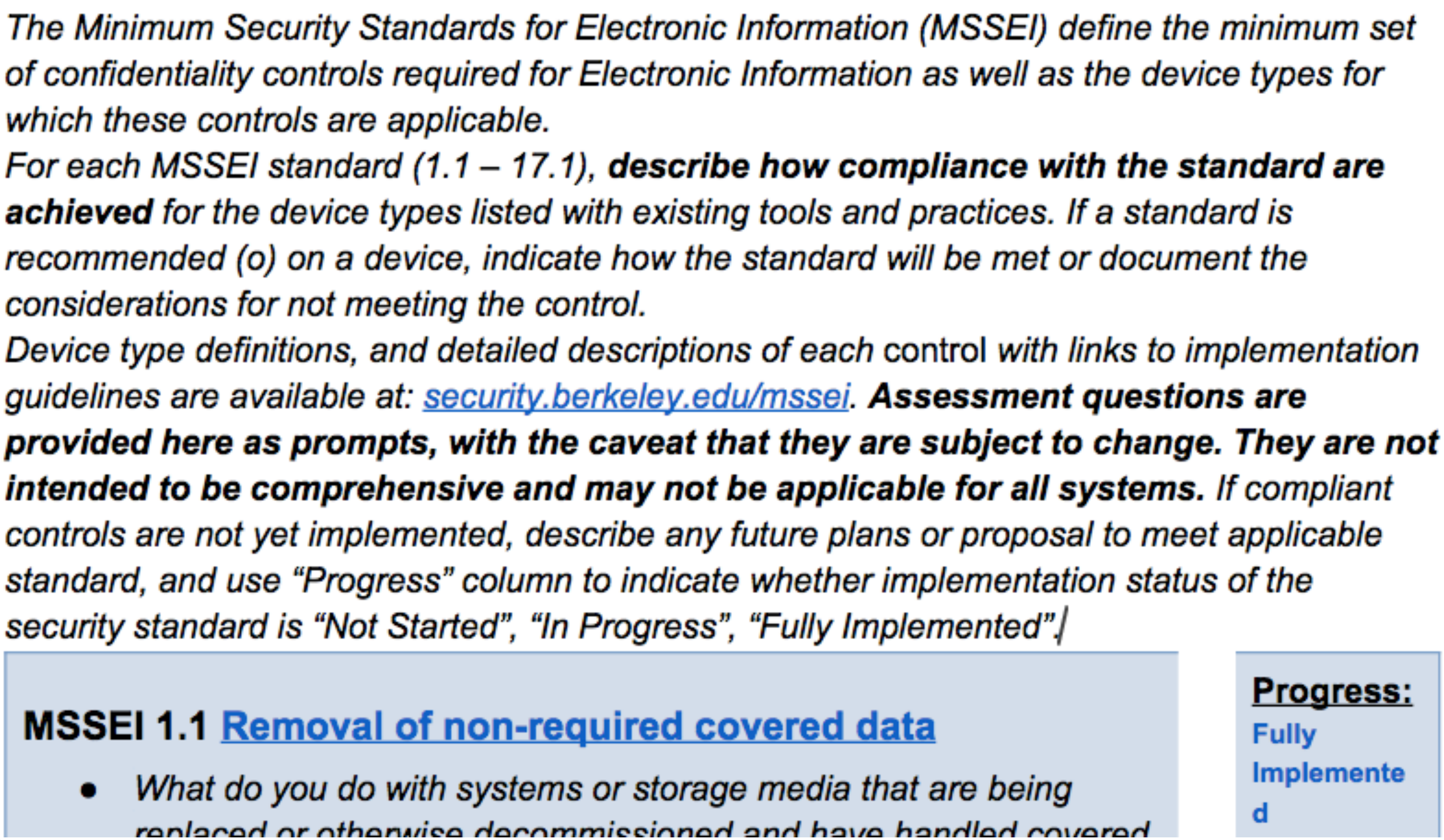
Please list any support and development staff that have elevated privileges in the application or its underlying systems, including their roles and responsibilities in supporting/developing this application.   In the responsibilities column, please make note if a role is temporary. Examples of temporary roles may include short-term contractors or support staff that will lose their elevated access to application in the near future (3 – 6 months). Elevated privileges in this case may mean permissions to change application configuration, bulk access to covered data, etc.



Name	Role	Application Responsibilities	Email Address
[redacted]	DevOps lead	permanent	[redacted]
[redacted]	IT Manager	permanent	[redacted]
[redacted]	Lead Developer	permanent	[redacted]
[redacted]	Release Manager	permanent	[redacted]

Step 6 -Meeting MSSEI Requirements

Derived from: https://www.sans.org/critical-security-controls/



The Minimum Security Standards for Electronic Information (MSSEI) define the minimum set of confidentiality controls required for Electronic Information as well as the device types for which these controls are applicable.

*For each MSSEI standard (1.1 – 17.1), **describe how compliance with the standard are achieved** for the device types listed with existing tools and practices. If a standard is recommended (o) on a device, indicate how the standard will be met or document the considerations for not meeting the control.*

*Device type definitions, and detailed descriptions of each control with links to implementation guidelines are available at: security.berkeley.edu/mssei. **Assessment questions are provided here as prompts, with the caveat that they are subject to change. They are not intended to be comprehensive and may not be applicable for all systems.** If compliant controls are not yet implemented, describe any future plans or proposal to meet applicable standard, and use "Progress" column to indicate whether implementation status of the security standard is "Not Started", "In Progress", "Fully Implemented".*

MSSEI 1.1 Removal of non-required covered data

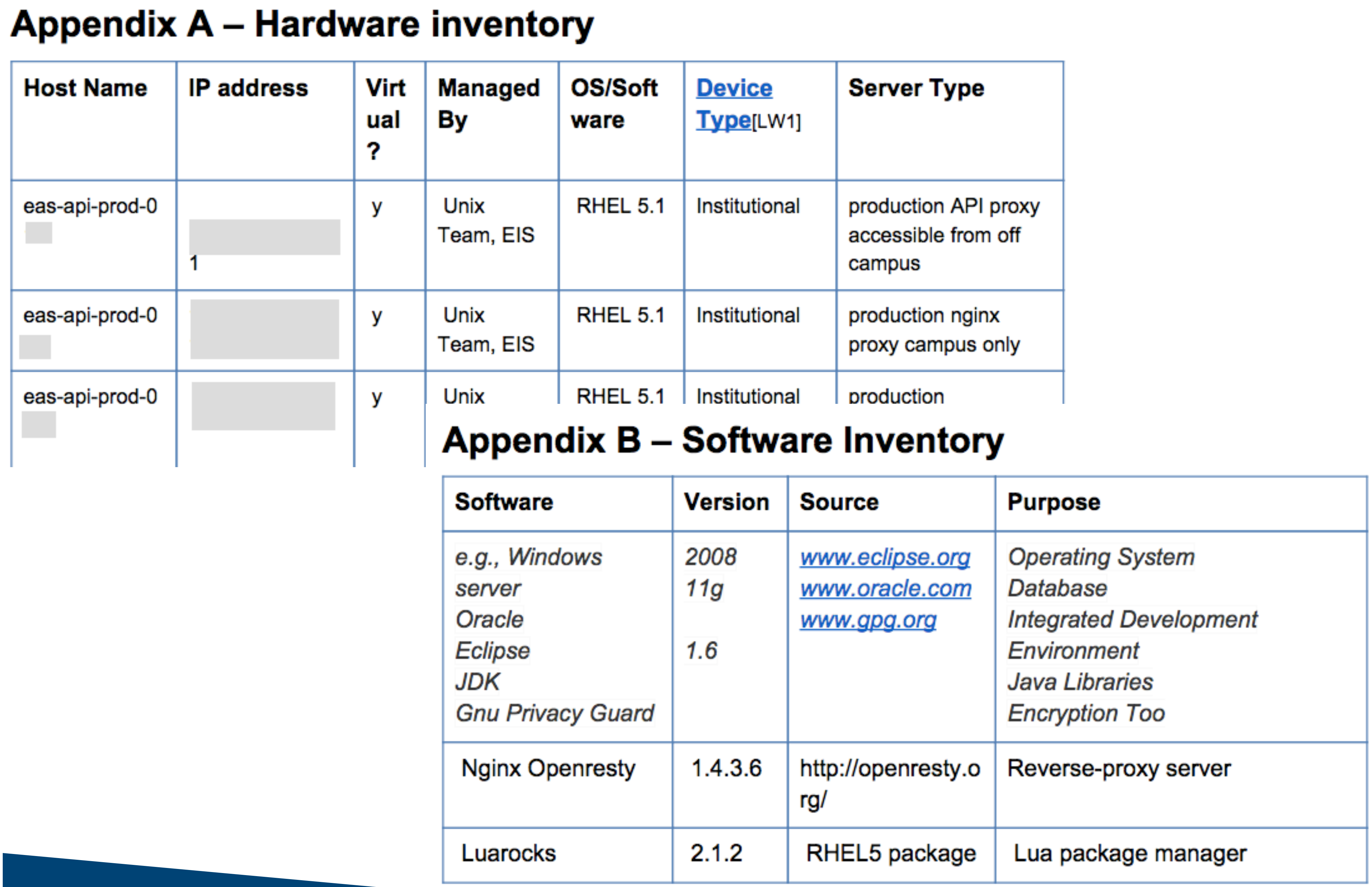
- *What do you do with systems or storage media that are being replaced or otherwise decommissioned and have handled covered*

Progress:

Fully Implemented

MSSEI Self Assessment Plan - High Level Requirements (small subset)

- Authenticated Scans
- Intrusion Detection
- Data flow and review
- Systems Inventory
- Build and Lifecycle
- Account Management
- “Hardware” Firewall
- Network Partitioning
- Audit Logging
- Encryption in Transit
- Secure Deletion



Appendix A – Hardware inventory

Host Name	IP address	Virtual?	Managed By	OS/Software	Device Type[LW1]	Server Type
eas-api-prod-0 [redacted]	[redacted] 1	y	Unix Team, EIS	RHEL 5.1	Institutional	production API proxy accessible from off campus
eas-api-prod-0 [redacted]	[redacted]	y	Unix Team, EIS	RHEL 5.1	Institutional	production nginx proxy campus only
eas-api-prod-0 [redacted]	[redacted]	y	Unix	RHEL 5.1	Institutional	production

Appendix B – Software Inventory

Software	Version	Source	Purpose
e.g., Windows server *Oracle* *Eclipse* *JDK* *Gnu Privacy Guard*	*2008* *11g* *1.6*	www.eclipse.org www.oracle.com www.gpg.org	*Operating System* *Database* *Integrated Development Environment* *Java Libraries* *Encryption Too*
Nginx Openresty	1.4.3.6	http://openresty.org/	Reverse-proxy server
Luarocks	2.1.2	RHEL5 package	Lua package manager

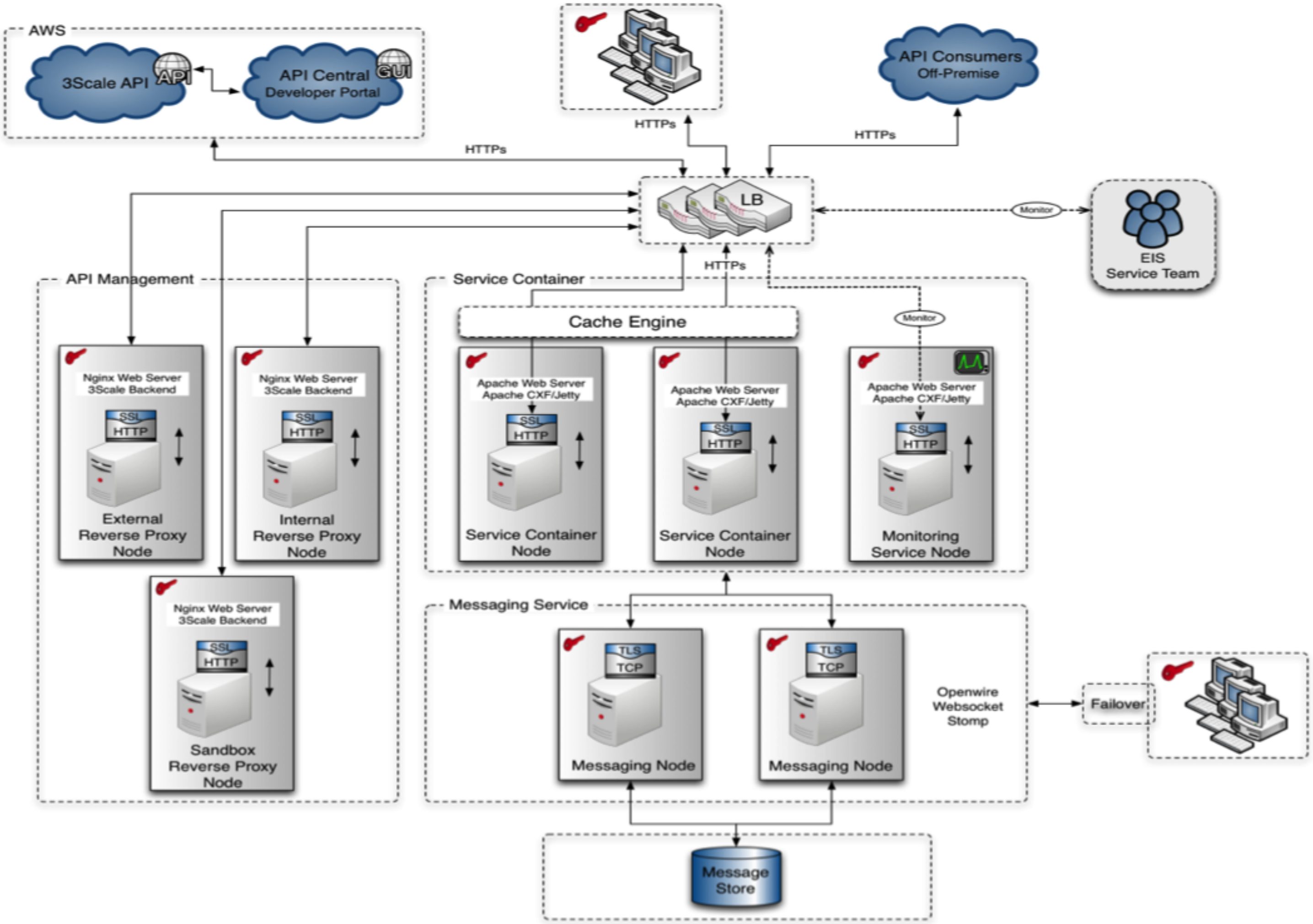
Berkeley
UNIVERSITY OF CALIFORNIA

Baseline

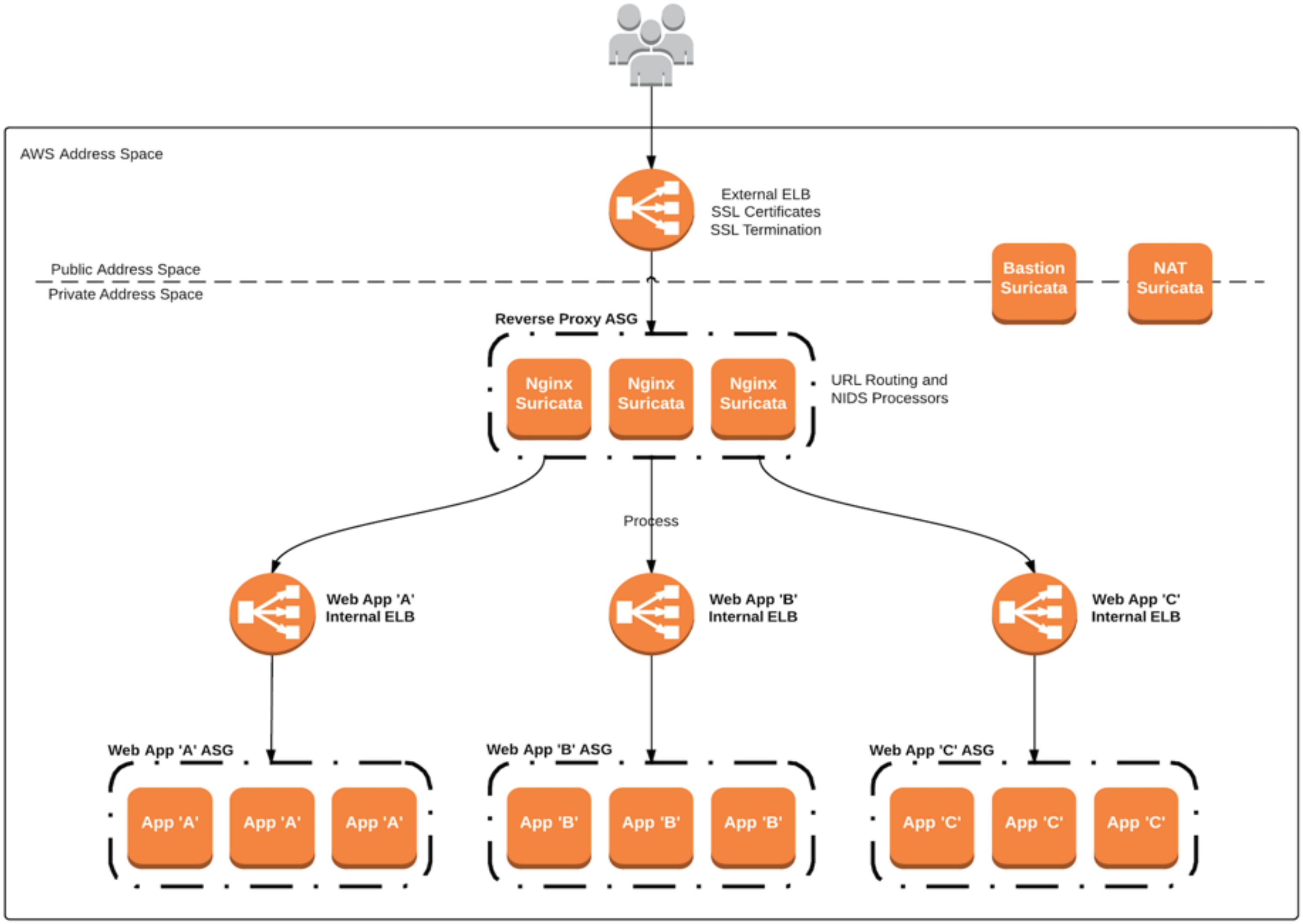
http://aws.amazon.com/whitepapers/aws-security-best-practices/

- 2 Factor authentication for AWS Console
- CF defined IAM Roles for all Instances
- Encryption for all comms in and out of VPCs
- Patching of security packages via yum-cron
- Identify credentials and their lifecycle
- Minimal (if any) data on EBS
- Minimal software deployment

API Service – On premises architecture

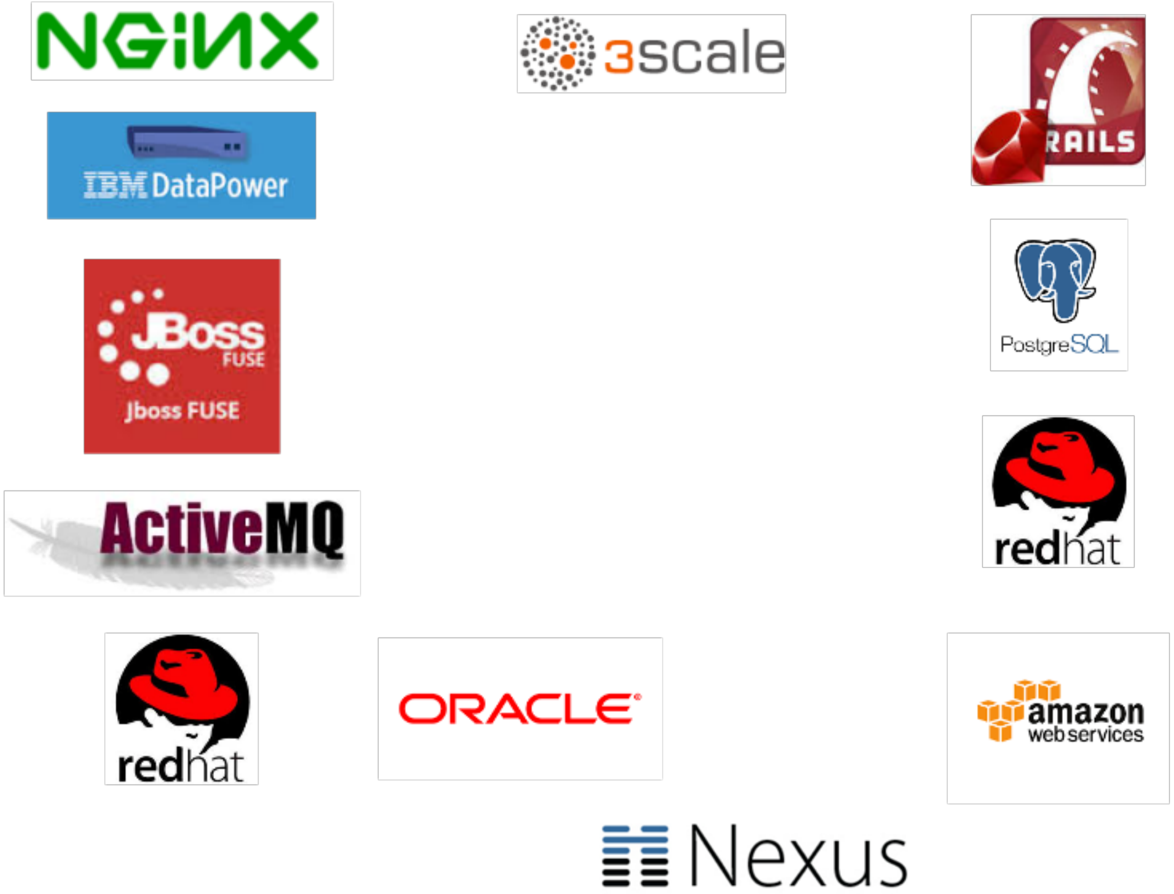


API Service – AWS architecture

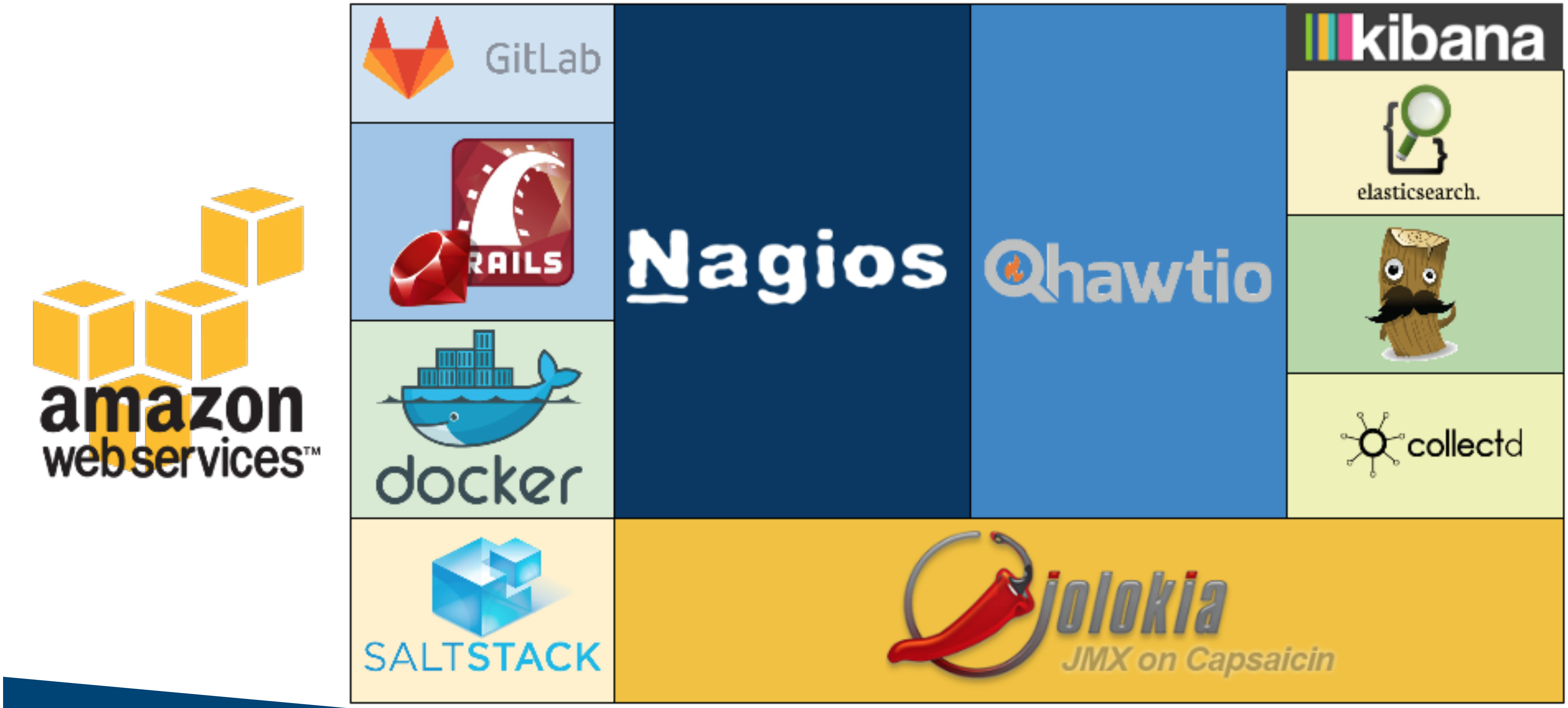


Berkeley
UNIVERSITY OF CALIFORNIA

API Service – Application stack



Management software stack



API Service – Security architecture

Suricata

Open Source IDS / IPS / NSM engine

http://suricata-ids.org/

AppScan

IBM. Rational.

EMERGING THREATS

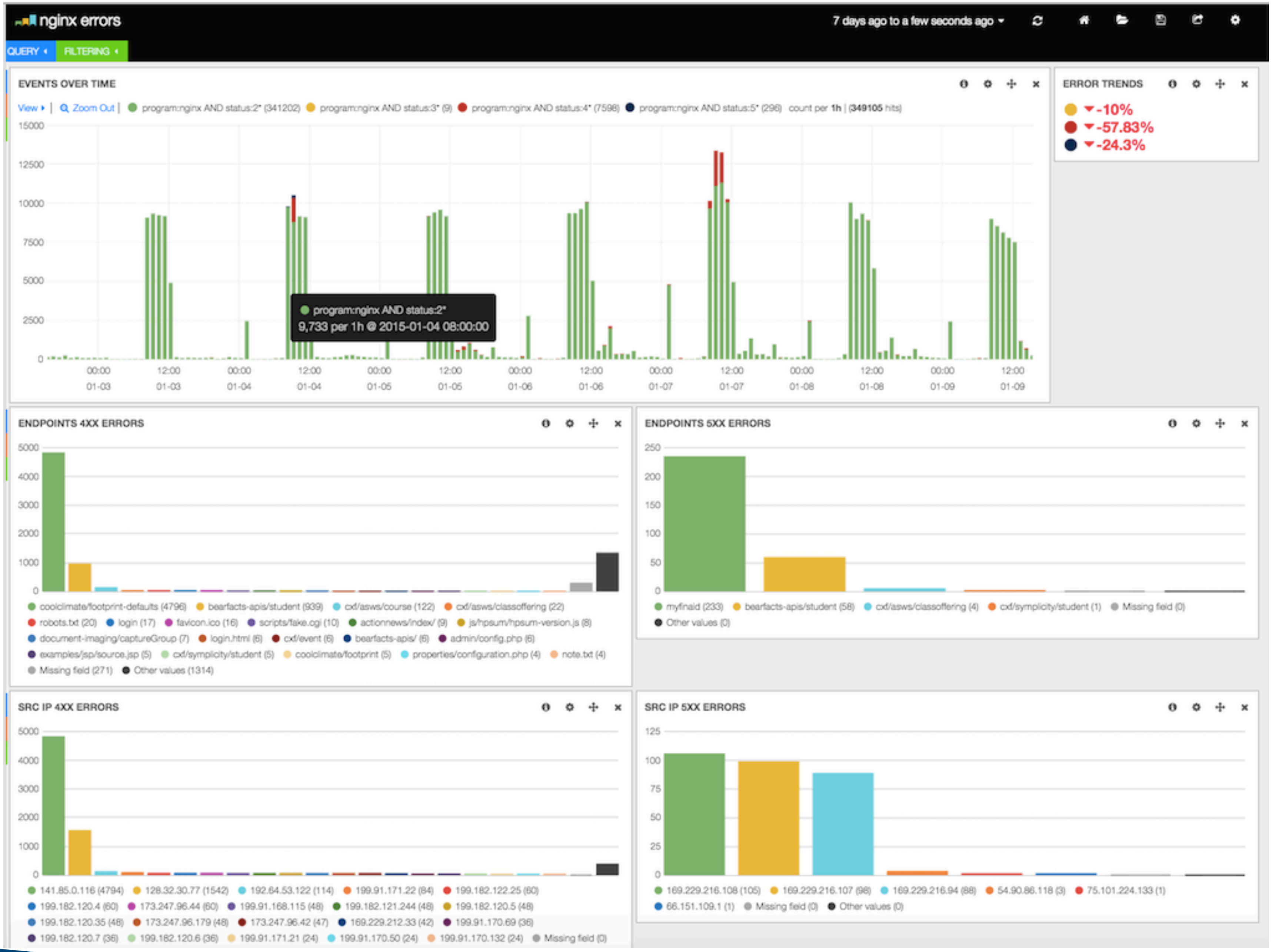
http://emergingthreats.net/products/etpro-ruleset/

Site24x7

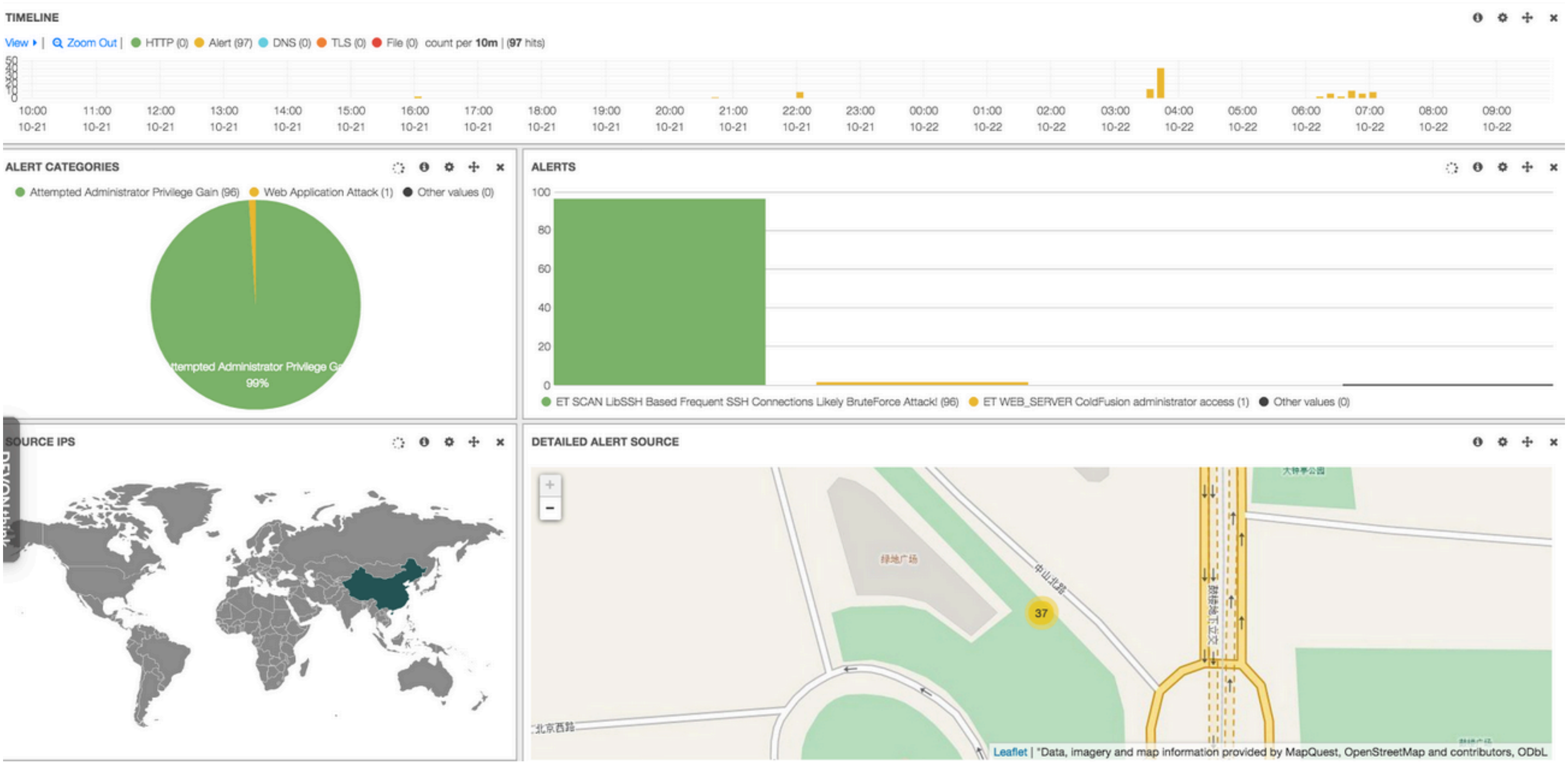
elasticsearch

http://www.elasticsearch.org/overview/kibana/

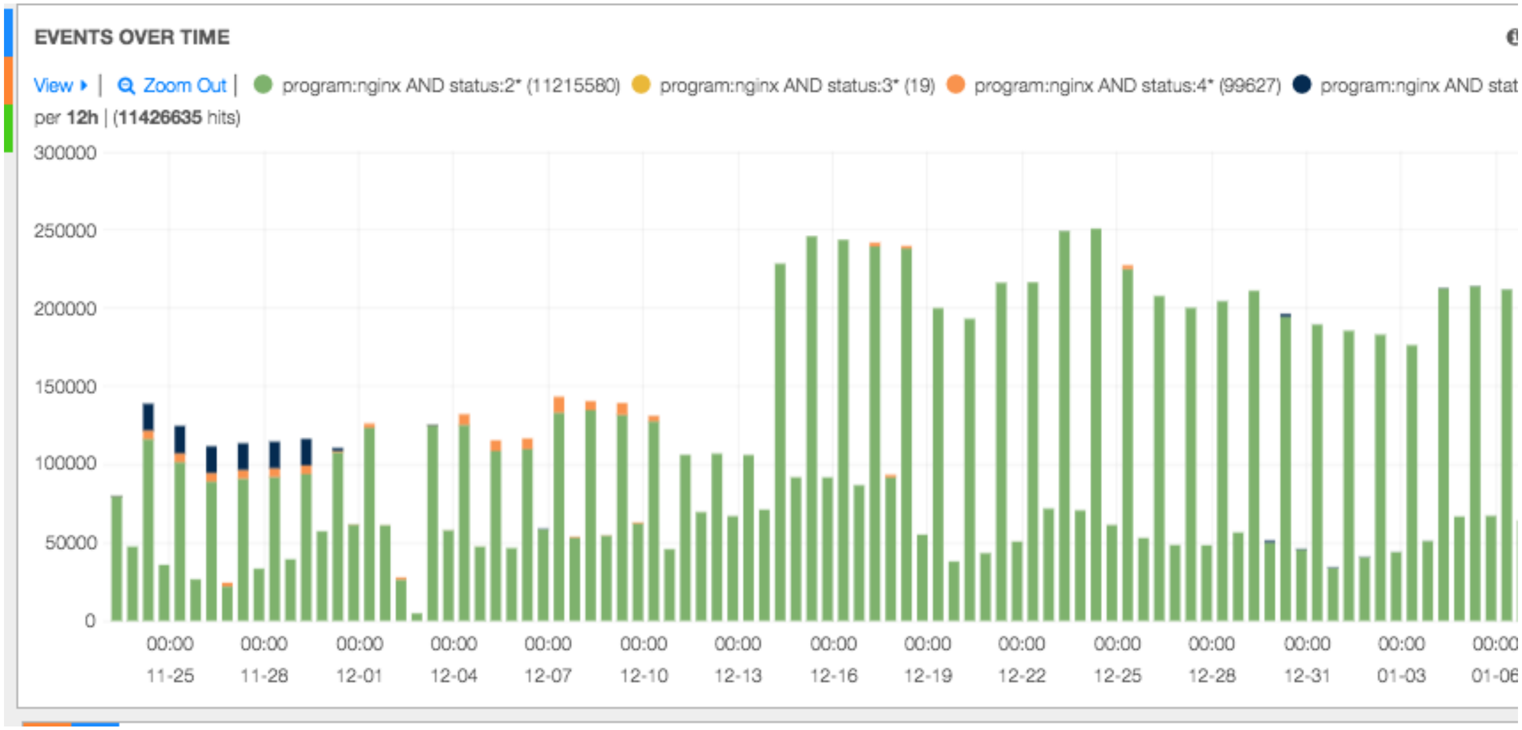
SALTSTACK



Berkeley
UNIVERSITY OF CALIFORNIA


TIMELINE
View ▸ | Zoom Out | ● HTTP (0) ● Alert (97) ● DNS (0) ● TLS (0) ● File (0) count per 10m | (97 hits)
50
40
30
20
10
0
10:00 10-21
11:00 10-21
12:00 10-21
13:00 10-21
14:00 10-21
15:00 10-21
16:00 10-21
17:00 10-21
18:00 10-21
19:00 10-21
20:00 10-21
21:00 10-21
22:00 10-21
23:00 10-21
00:00 10-22
01:00 10-22
02:00 10-22
03:00 10-22
04:00 10-22
05:00 10-22
06:00 10-22
07:00 10-22
08:00 10-22
09:00 10-22
ALERT CATEGORIES
● Attempted Administrator Privilege Gain (96) ● Web Application Attack (1) ● Other values (0)
ttempted Administrator Privilege G
99%
ALERTS
100
80
60
40
20
0
● ET SCAN LibSSH Based Frequent SSH Connections Likely BruteForce Attack! (96) ● ET WEB_SERVER ColdFusion administrator access (1) ● Other values (0)
SOURCE IPS
DETAILED ALERT SOURCE
+
−
绿地广场
中山北路
37
北京西路
Leaflet | "Data, imagery and map information provided by MapQuest, OpenStreetMap and contributors, ODbL

SIS: This is what prelaunch load test results looked like:

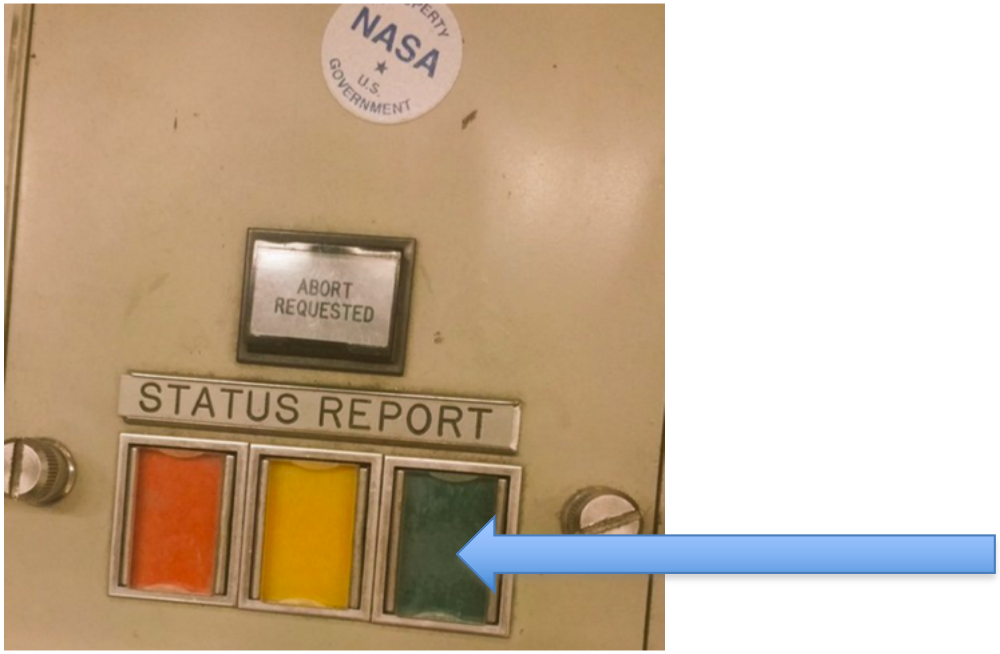


EVENTS OVER TIME

View ▸ | Zoom Out | ● program:nginx AND status:2* (11215580) ● program:nginx AND status:3* (19) ● program:nginx AND status:4* (99627) ● program:nginx AND stat

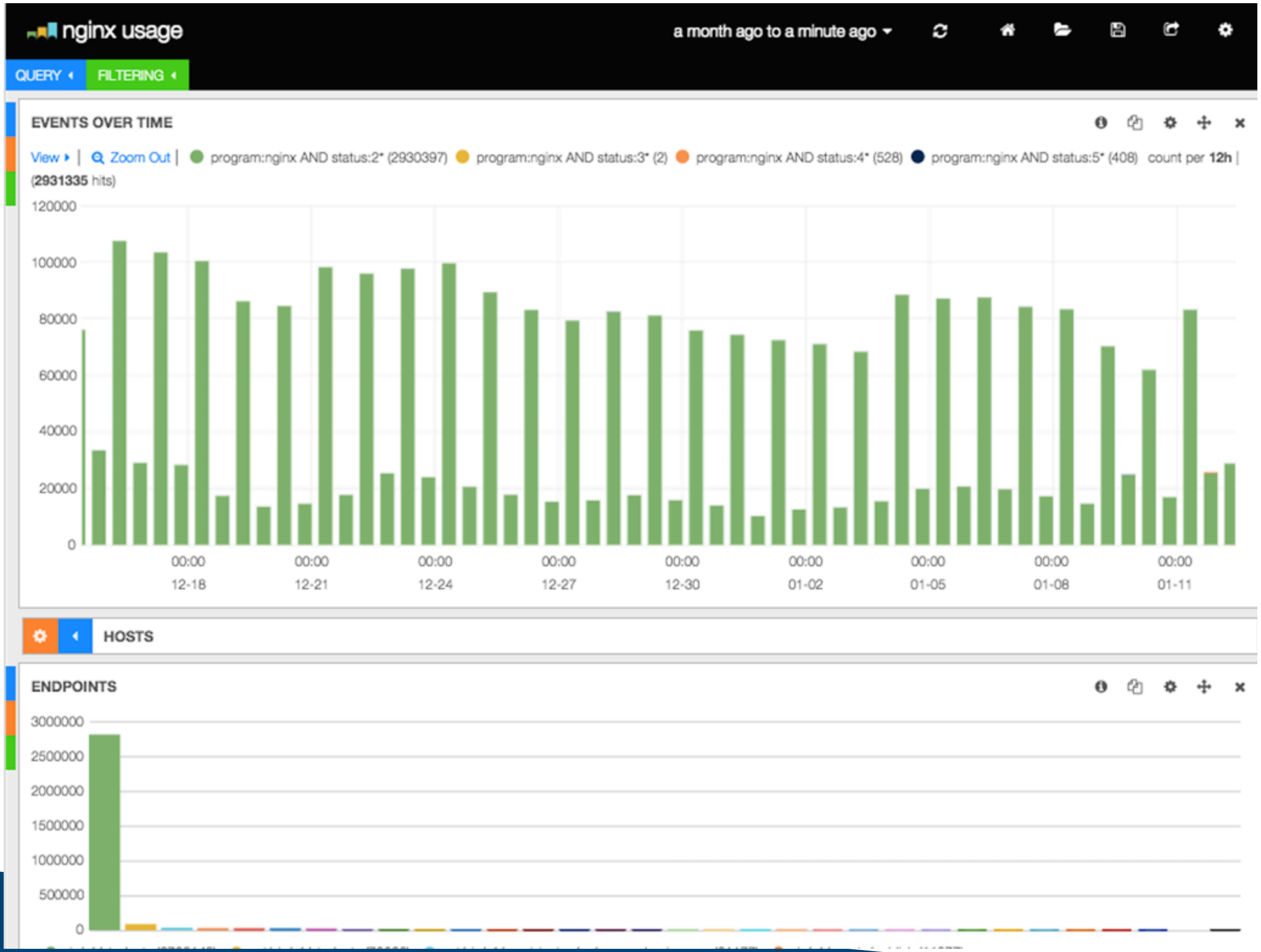
per 12h | (11426635 hits)

Going live with APIs for PeopleSoft Campus Solutions



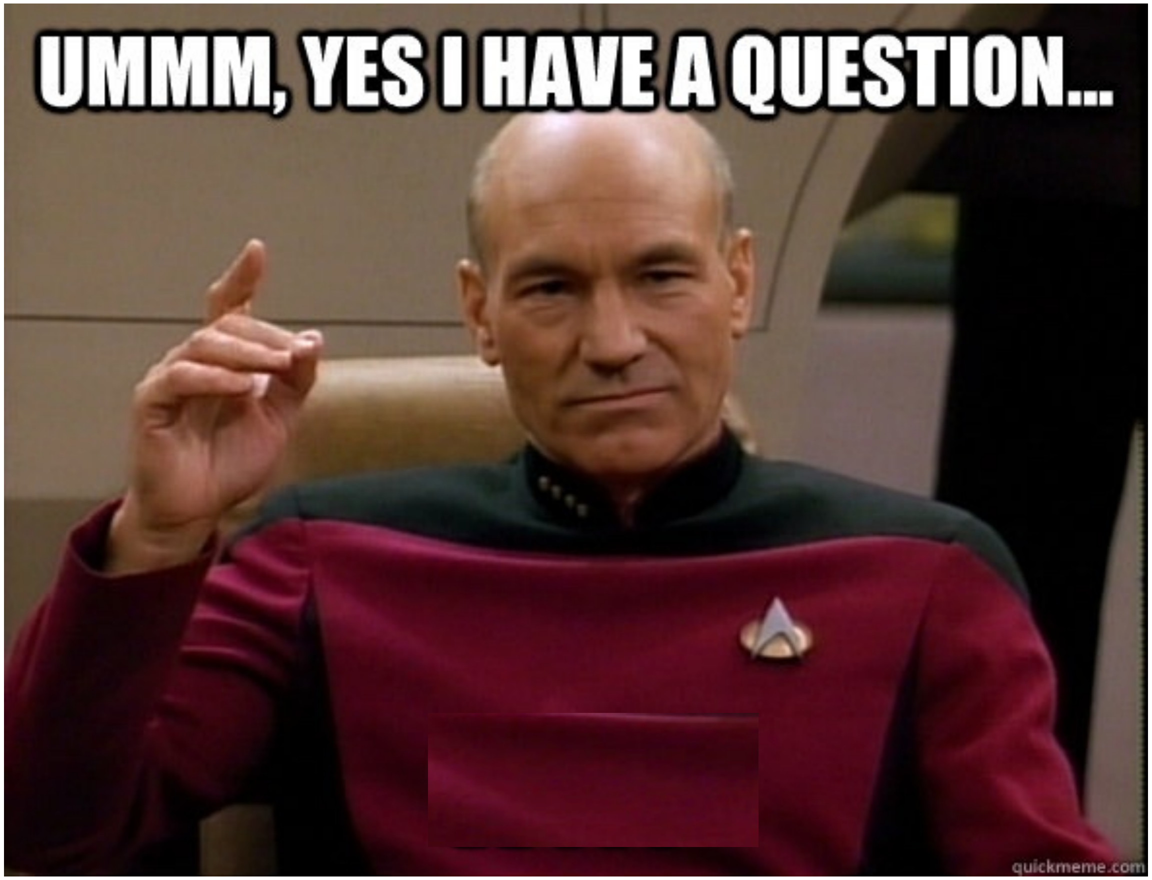
Berkeley
UNIVERSITY OF CALIFORNIA

Day/week one looked like:



● program:nginx AND status:2* (2930397) ● program:nginx AND status:3* (2) ● program:nginx AND status:4* (528) ● program:nginx AND status:5* (408) count per **12h** | (**2931335** hits)

Questions?



Thank you to the UCB IT Security Team and the IST-API Integration Team