

# Systems Engineering Designs on the Cloud

**Daniel Hettema**, Chris Ritter, and Steven H. Dam,  
Ph.D., ESEP, President, SPEC Innovations,  
571-485-7799  
daniel.hettema@specinnovations.com

October 2013

# Overview

- What is the Problem?
- What Does Scalability Mean?
- What is Cloud Computing?
- What Does a Commercial Cloud Vendor (Google) Do?
- How Can We Design Software for Scalability?
- Summary

# What Is the Problem?

*“Engineers are engaged today in the design and development of large - scale, interconnected systems of staggering complexity and criticality.”  
Mr. Stephen P. Welby, Deputy Assistant Secretary of Defense for Systems Engineering*

– from INCOSE International Symposium, 6/25/2013

- The systems we build today have millions of parts, distributed worldwide
- They are all interconnected
- Teams of designers, developers, integrators, testers, and operators all take pieces of the problem and try to bring them together in different information environments in their own stovepipes

***The environments of today limit the amount of information that can be captured and shared, because they do not scale to the need***



# What Does Scalability Mean?

- Many systems and products claim to scale, but what does it mean?
- If the tool and product are limited to a standard client server environment, then the hardware will limit their ability to scale
- But worse, unless the system is designed to scale, the user interface will become so slow and cumbersome that finding anything will be very difficult to impossible

Scalable: “capable of being easily expanded or upgraded on demand”

From <http://www.merriam-webster.com/dictionary/scalable>

*Can you find what you are looking for on your “P” drive?*

# What is Cloud Computing?

Hint: It's not just a website

# What is cloud computing?

- Definition from NIST:
  - *Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources 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***

From presentation by Jim Sweeney, GTSI at the Technology Leadership Series 2012 Seminar, January 19, 2012



# Five Essential Characteristics

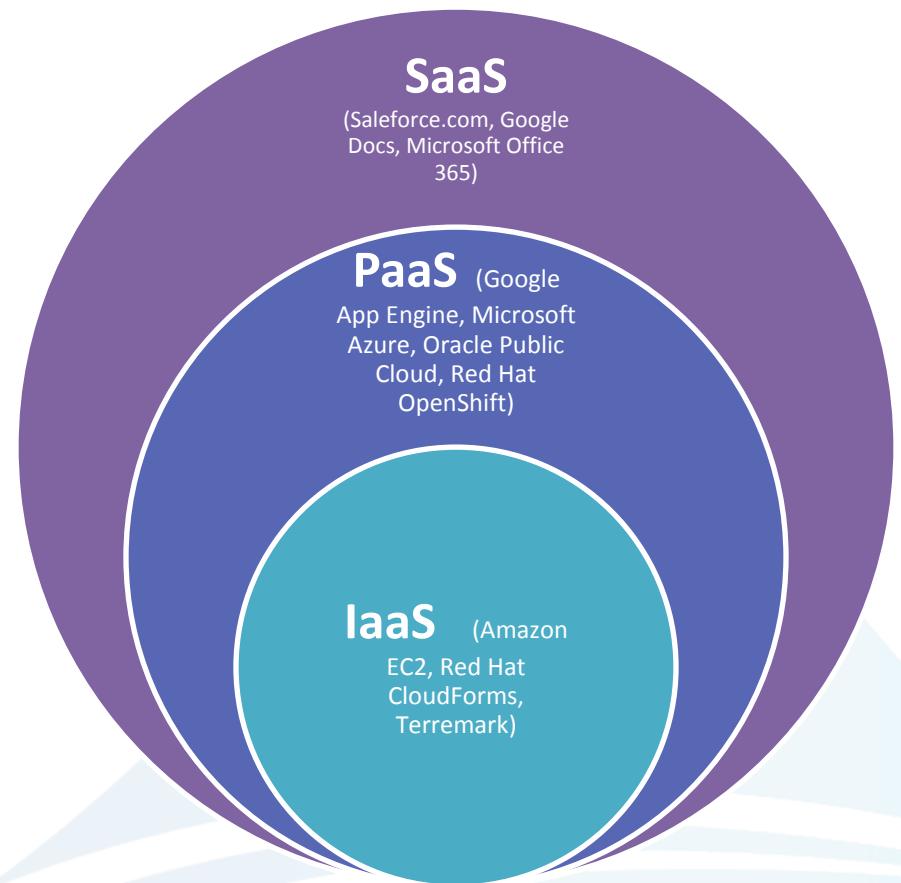
- ***On-demand self-service.*** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- ***Broad network access.*** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- ***Resource pooling.*** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
- ***Rapid elasticity.*** Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- ***Measured Service.*** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

From presentation by Jim Sweeney, GTSI at the Technology Leadership Series 2012 Seminar, January 19, 2012



# Three Service Models

- **Software as a Service (SaaS):** The end user system
- **Platform as a Service (PaaS):** Tools and services to create a SaaS Application
- **Infrastructure as a Service (IaaS):** Full control of the software stack and services





# Four Deployment Models

## *Private cloud*

- Operated solely for an organization
- May be managed by the organization or a third party

## *Public cloud*

- Available to the general public or a large industry group
- Owned by an organization selling cloud services

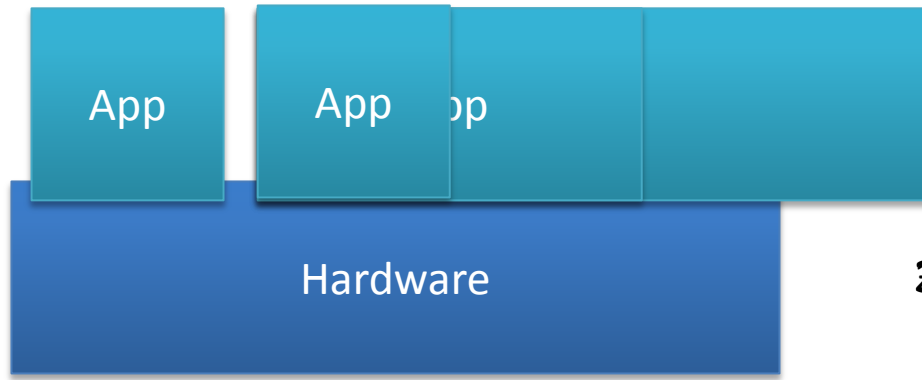
## *Community cloud*

- Shared by several organizations
- Managed by the organizations or a third party

## *Hybrid cloud*

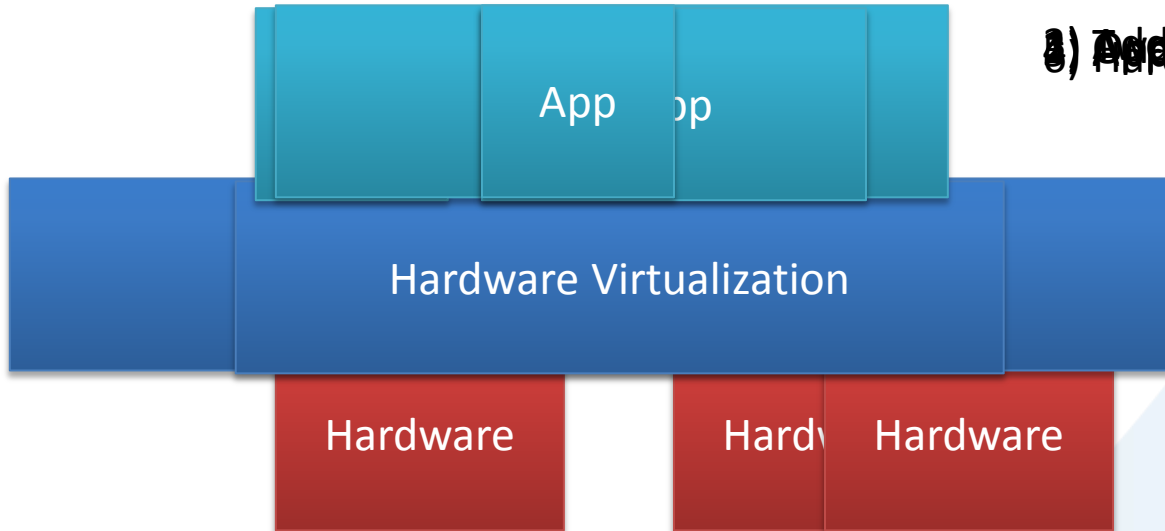
- composition of two or more clouds that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability

# Normal Server Deployment



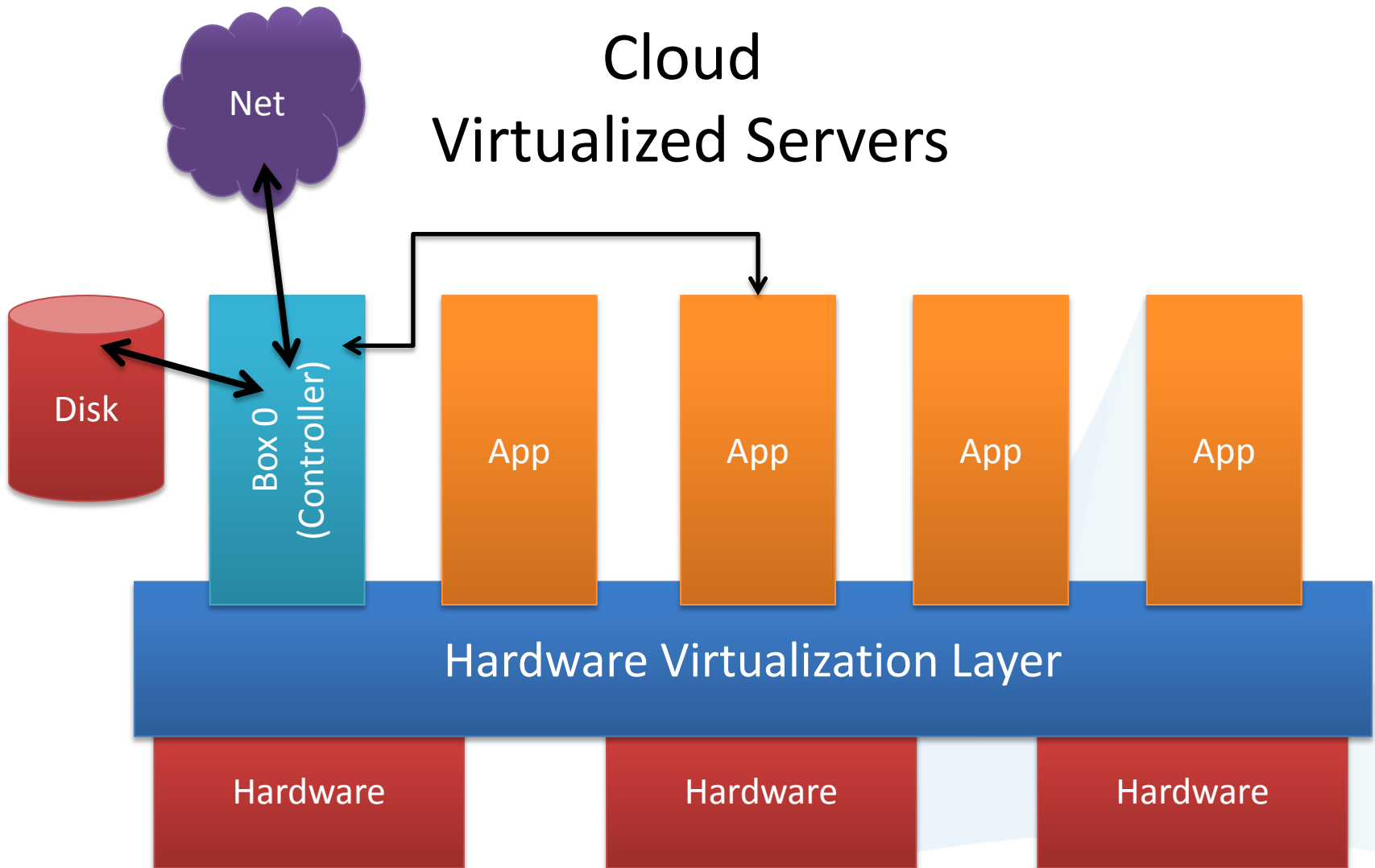
2) ~~One application, but multiple instances~~  
normal conditions

# Virtualized Server Deployment



1) Applications are not installed on hardware server machines, but are virtualized conditions.

# Cloud Virtualized Servers

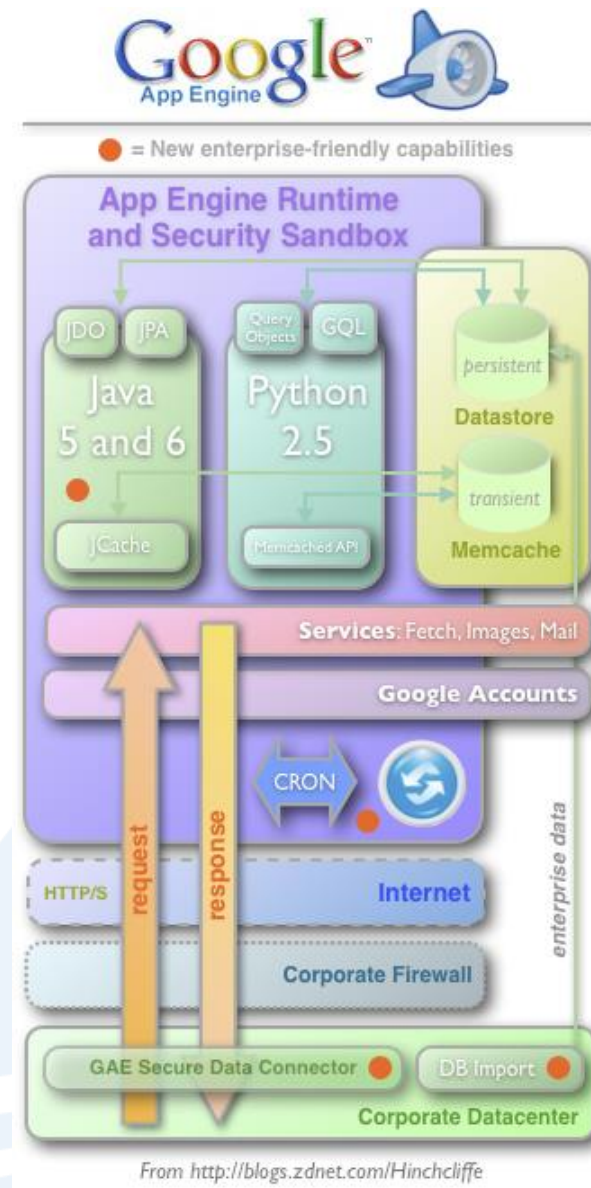


# What Does a Commercial Cloud Vendor (Google) Do?

A leader in cloud computing

# Google AppEngine

- PaaS
- Supports Java, Python, Go and PHP
- Google Cloud SQL
- Memcache
- Task Queues
- Auto-scaling



# World-wide Data Centers

## Data center locations

We own and operate data centers around the world to keep our products running 24 hours a day, 7 days a week. Find out more about our data center locations, community involvement, and [job opportunities](#) in our locations around the world.

### Americas

Berkeley County, South Carolina  
Council Bluffs, Iowa  
Douglas County, Georgia  
Quilicura, Chile  
Mayes County, Oklahoma  
Lenoir, North Carolina  
The Dalles, Oregon

### Asia

Hong Kong  
Singapore  
Taiwan

### Europe

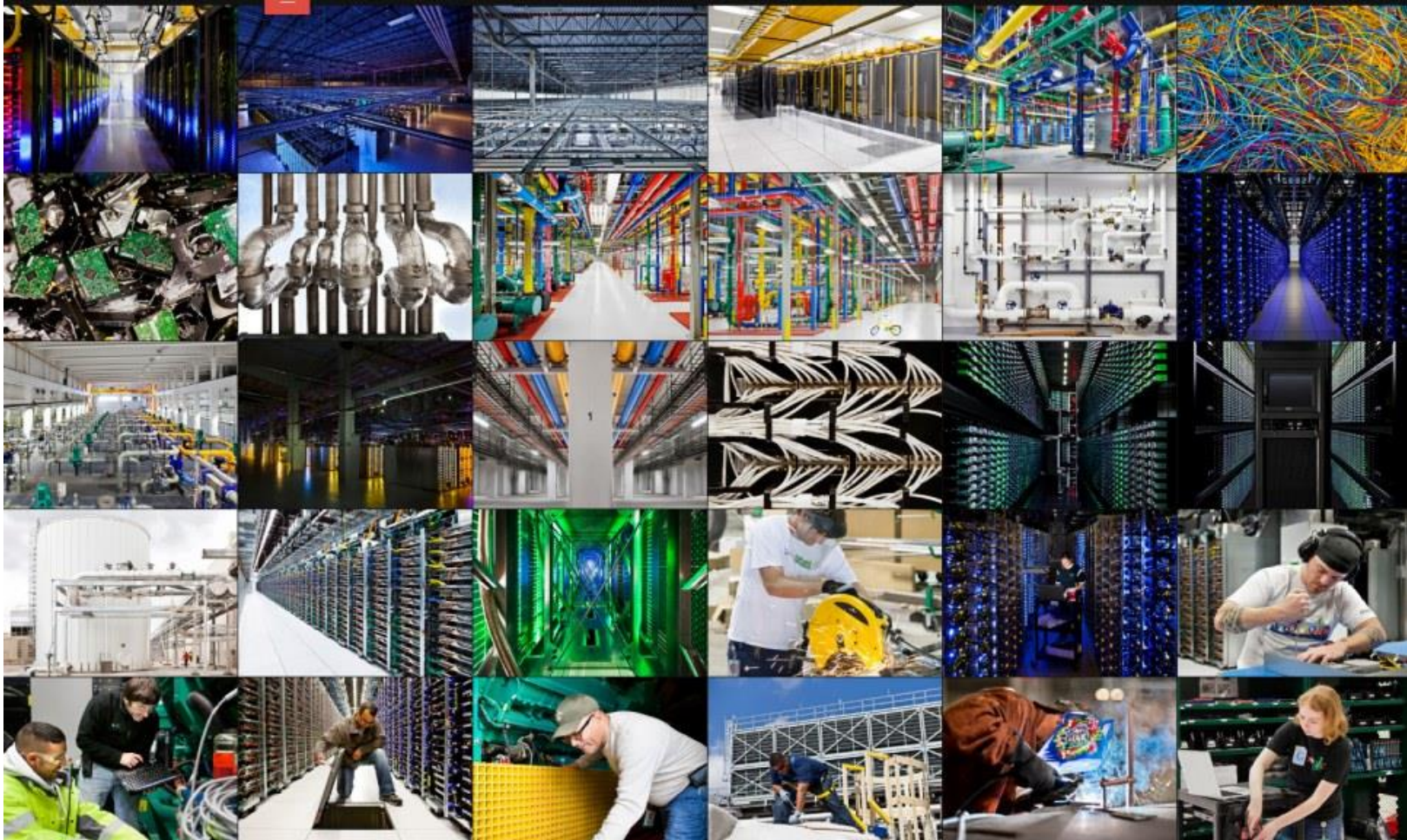
Hamina, Finland  
St Ghislain, Belgium  
Dublin, Ireland



*Regions can be specified*



# Google Data Centers





# Google Apps

- Google Apps for Government available

- FISMA accredited
- Automatic backups
- 24x7x365 network operations
- Your data is yours (“When you put your data in Google Apps, you still own it, and it says just that in our contracts. We don’t scan your data in order to show you ads.”)

- Designed to scale



# How Can We Design Software for Scalability?

# Designing for Scalability

- To state the obvious: you must take scalability into account all through out the design process
  - User interface
  - Databases
  - Algorithms
- We encountered these problems as we developed Innoslate®

# Requirements View

*Automated Quality Check requires careful algorithm development for large numbers of requirements*

**Quality Check**

Statement	Quality Score	Labels	Clear
NGO.1 Provide timely detection Provide timely detection and notification of potentially dangerous wildfires	N/A	Baseline	N/A
NGO.1.1 Detect in less than a day Detect a potentially dangerous wildfire in less than 1 day (threshold), 12 hours (objective)	N/A	Baseline FireSAT - Requr	N/A
NGO.1.1.1 Detection The FireSAT system shall detect potentially dangerous wildfires (defined to be greater than 150 m in any linear dimension) with a confidence interval of 95%. Rationale: The USFS has determined that a 95% confidence interval is sufficient for the scope of FireSAT.	100%	Baseline FireSAT - Requr Functional Requr	Yes
NGO.1.1.2 Coverage The FireSAT system shall cover the entire United States, including Alaska and Hawaii. Rationale: For a US Government funded program, coverage of all 50 states is a political necessity.	44%	Baseline FireSAT - Requr Functional Requr	No Contains a conjunction; consider separating into two statements.
NGO.1.2 Provide notification within 1 hour Provide notification to USFS within 1 hour of detection (threshold), 30 minutes (objective)	N/A	Baseline FireSAT - Requr	N/A
NGO.2.0 Provide continuous monitoring Provide continuous monitoring of dangerous and potentially dangerous wildfires	N/A	Baseline FireSAT - Requr	N/A
NGO.2.1 Provide 24/7 monitoring Provide 24/7 monitoring of high priority dangerous and potentially dangerous wildfires	N/A	Baseline FireSAT - Requr	N/A
NGO.2.1.1 Orbit Altitude FireSAT satellites shall maintain an orbit altitude of 700 km	22%	Baseline FireSAT - Requr Performance Re	Yes
NGO.2.1.2 Number of Satellites The FireSAT system shall include 2 satellites in one plane separated 180 degrees in true anomaly	N/A	Baseline FireSAT - Requr	N/A
NGO.3.0 Reduce the economic impact Reduce the economic impact of wildfires	N/A	Baseline FireSAT - Requr	N/A
NGO.3.1 Reduce cost Reduce the average annual cost of fighting wildfires by 20% from 2006 average baseline	N/A	Baseline FireSAT - Requr	N/A
NGO.3.2 Reduce property loss Reduce the annual property losses due to wildfires by 25% over 2006 baseline	N/A	Baseline FireSAT - Requr	N/A
NGO.4.0 Reduce the risk Reduce the risk to firefighting personnel	N/A	Baseline FireSAT - Requr	N/A
NGO.4.1 Reduce fire size at first contact Reduce the average size of fire at first contact by firefighters by 20% from 2006 average baseline	N/A	Baseline FireSAT - Requr	N/A
NGO.4.2 90% user satisfaction	N/A	Baseline	N/A

**Labels**

- Classification: Unclas...
- Classifications
- CONOPS Fmwks
- CONOPS: FireSAT
- Designators
- Environmental Requir...
- Functional Requirement**
- Identified Risk
- Impacted Process or ...
- Impacted Skill or Co...
- Interface Requirement
- ISoC: Impact
- ISoC: Justification
- Key Component
- Key Policy
- Master Organization
- Performance Requir...
- Reliability Requirement
- Risk Labels
- Rpt Fmwks
- RskLvlAr: High
- RskLvlAr: Low

**Use of labels instead of folders for organizing information**

# Database View

*Sharing databases worldwide requires careful design to deal with large numbers of contributors*

The screenshot shows a web-based database management system. The top navigation bar includes 'MENU', 'Database', 'Requirements', and 'Search'. A red circle highlights the 'Search' button, with a red arrow pointing to it from the text 'Search capability requires indexing to speed up for large databases'. Another red circle highlights the 'Share' button in the top right corner, with a red arrow pointing to it from the text 'Sharing databases worldwide requires careful design to deal with large numbers of contributors'. The main content area displays a list of database entries, each with a title, description, and a 'Share' button. A red circle highlights a chat window in the bottom right corner, with a red arrow pointing to it from the text 'Built-in Chat has scalability issues associated with it'. The chat window shows a conversation between 'Michael Campbell' and 'Online Users'.

*Search capability requires indexing to speed up for large databases*

*Built-in Chat has scalability issues associated with it*

*Designed to scale to large data sets*



# Entity View

*Detailed history of changes available for each element can become a scaling issue*

*Tabs for logical grouping of relationships reduces information overload*

The screenshot displays the 'Entity View' for 'FireSAT System'. The interface includes a top navigation bar with 'Database', 'Requirements', and 'Search'. Below this is a toolbar with 'Save', 'Diagrams', 'History', 'Duplicate', 'More', and 'Delete'. The main content area is divided into 'Attributes' (Name: FireSAT System, Number: FS, Description: The FireSAT system detects wildfires before they become a major problem and notifies firefighting personnel as to the location, direction and speed of the fire to enable more rapid response to putting out the fire. This system will also enable the public to have better, more timely notification to reduce risk and cost of evacuations.), 'Comments' (Steven Dam posted on 9/18/2013: Derived from the Applied Space Systems Engineering book.), and 'Relationships' (Popular, Program Management, Locations, All). A sidebar on the left shows a globe image and a list of labels including Actor, Architecture, CONOPS Fmwkrs, CONOPS: FireSAT, Classification: Unclassified, Classifications, Context, Designators, and Entity. A 'History' button in the toolbar is circled in red. A tooltip is visible over the 'Description' field in the Relationships section.

*Capture and storage of pictures requires scalability design*

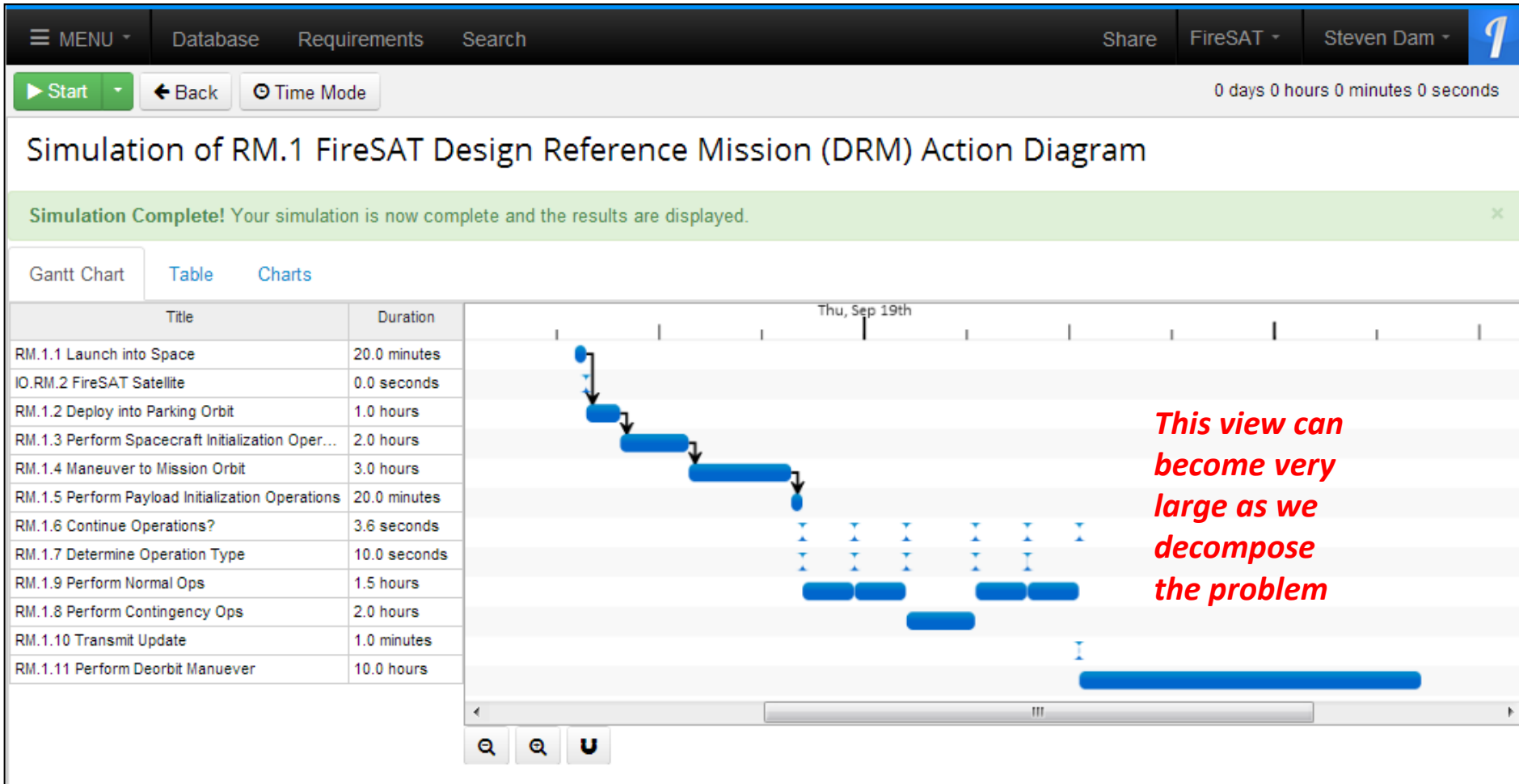
*Capturing comments, like in a blog, has scaling issues*

# Action Modeling (Functional View)

The screenshot shows a software interface for Action Modeling (Functional View). The top menu bar includes 'MENU', 'Database', 'Requirements', and 'Search'. The toolbar contains 'Save', 'Download as', 'Simulate', and 'Auto Layout'. On the left, there is a 'New Existing' section with a grid of icons: 'Action', 'IO Input/Output', 'Parallel', 'OR Or', 'LOOP Loop', and 'SYNC Sync'. A red circle highlights the 'LOOP' and 'OR' icons, with a red arrow pointing to a text box that reads: "Design for speed in using a browser becomes a real challenge – even when not being concerned with scalability". Below the icons is a sidebar titled "RM.1 FireSAT Design Reference Mission (DRM)" with a description of the mission. The main area displays a flowchart diagram for the FireSAT mission. The flow starts at 'START' and goes through 'RM.1.1 Launch into Space', 'RM.1.2 Deploy into Parking Orbit', 'RM.1.3 Perform Spacecraft Initialization', 'RM.1.4 Maneuver to Mission Orbit', and 'RM.1.5 Perform Payload Initialization'. From 'RM.1.5', the flow goes to 'RM.1.6 Continue Operations?' which is a 'LOOP' icon. From 'RM.1.6', the flow goes to 'RM.1.7 Determine Operation Type' which is an 'OR' icon. From 'RM.1.7', the flow branches into 'Contingency' leading to 'RM.1.8 Perform Contingency Ops' and 'Normal' leading to 'RM.1.9 Perform Normal Ops'. Both 'RM.1.8' and 'RM.1.9' lead to 'RM.1.10 Transmit Update'. From 'RM.1.10', the flow goes to 'RM.1.11 Perform Deorbit Manuever' and then to 'END'. There are also dashed arrows from 'RM.1.1' to 'FireSAT Satellite' and 'Expanded Booster', and from 'RM.1.10' to 'Platform Telemetry', 'Payload Telemetry', and 'Payload Data'. A red circle highlights the 'LOOP' and 'OR' icons in the menu, with a red arrow pointing to a text box that reads: "Design for speed in using a browser becomes a real challenge – even when not being concerned with scalability".

*Action Diagrams for functional modeling designed to work on tablets, such as the iPad*

# Discrete Event Simulation



***Results saved automatically as an Artifact;  
Monte Carlo available in Professional Edition***





# Summary

# Summary

- Cloud computing is here
  - DoD needs to get fully on-board and leverage the commercial resources, not just re-invent the wheel
- Partitioning becomes less necessary and can now be done for organizational, rather than tool-related reasons
- Do not expect your desktop tools to port over to a cloud environment and scale
- Include scalability in the designs of all the software you build for the future