

Cloud Operating Systems

Vasia Kalavri (vasia@ac.upc.edu)

Universitat Politècnica de Catalunya

EMDC - EEDC 2011

Outline

- ★ Introduction
- ★ Cloud Operating Systems
 - ★ The Datacenter view
 - ★ Motivation
 - ★ Requirements
 - ★ Challenges
 - ★ Example: vSphere
 - ★ The Web view
 - ★ Current situation
 - ★ Goals & Obstacles
 - ★ Example: Gazelle
- ★ Conclusions

Outline

★ Introduction

★ Cloud Operating Systems

★ The Datacenter view

- ★ Motivation

- ★ Requirements

- ★ Challenges

- ★ Example: vSphere

★ The Web view

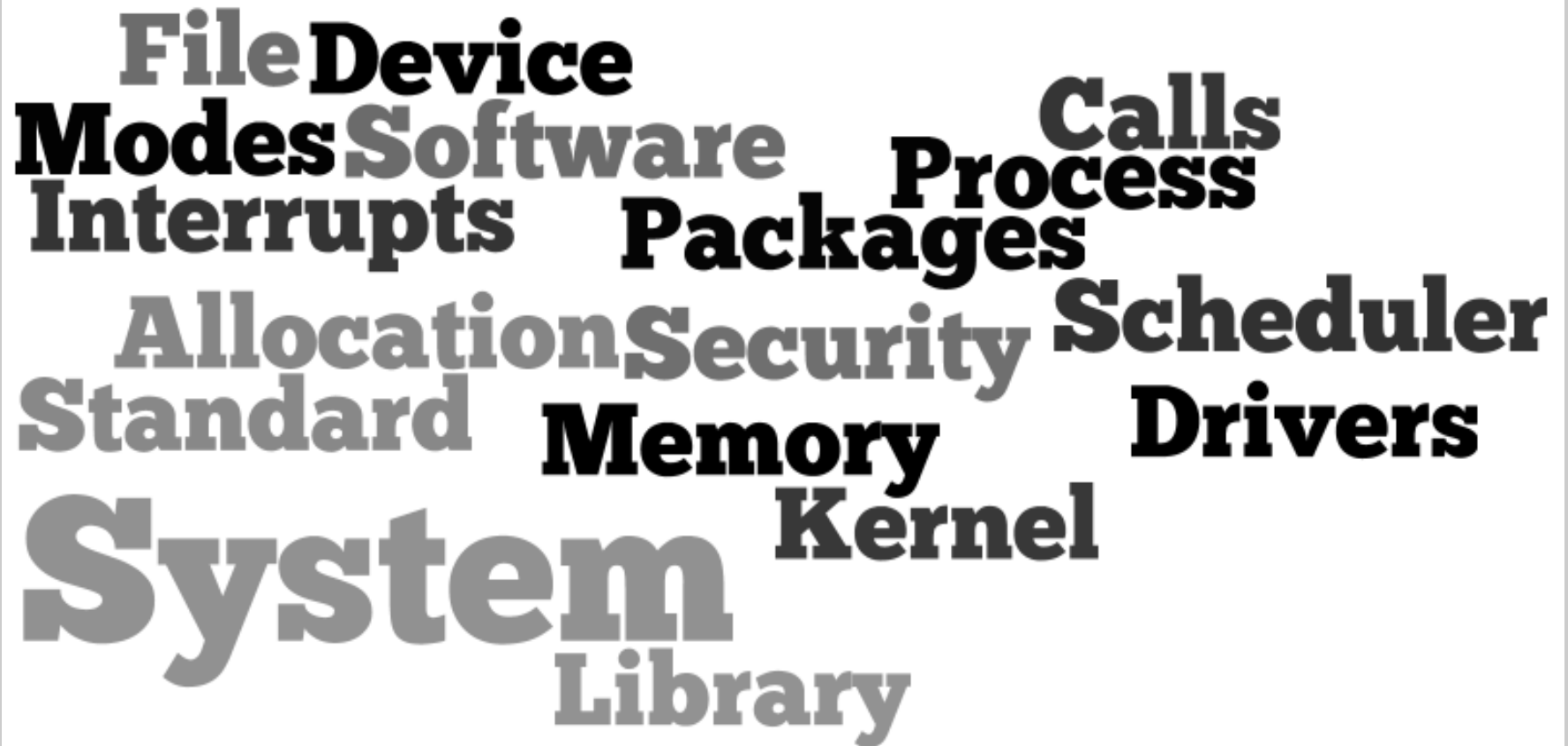
- ★ Current situation

- ★ Goals & Obstacles

- ★ Example: Gazelle

★ Conclusions

What is a Cloud Operating System?



A word cloud of operating system components. The words are arranged in a non-uniform, overlapping manner. The word 'System' is the largest and most prominent, located at the bottom left. Other large words include 'Scheduler', 'Process', 'Memory', 'Security', 'Allocation', 'Standard', 'Drivers', 'Kernel', 'Packages', 'Interrupts', 'Modes', 'File Device', 'Calls', and 'Library'. The words are in various shades of gray, with some being darker than others.

File Device
Modes **Software** **Process** **Calls**
Interrupts **Packages**
Allocation **Security** **Scheduler**
Standard **Memory** **Drivers**
System **Kernel**
Library

Definitions

- ★ “A software that manages **large** collections of infrastructure as a **seamless, flexible** and **dynamic** operating environment”
- ★ “An environment created in a user's machine from an **on-line** application stored on the **Cloud** and run through a Web **browser**”

Outline

- ★ Introduction

- ★ **Cloud Operating Systems**

 - ★ The Datacenter view

 - ★ Motivation

 - ★ Requirements

 - ★ Challenges

 - ★ Example: vSphere

 - ★ The Web view

 - ★ Current situation

 - ★ Goals & Obstacles

 - ★ Example: Gazelle

- ★ Conclusions

Outline

- ★ Introduction

- ★ **Cloud Operating Systems**

 - ★ **The Datacenter view**

 - ★ Motivation

 - ★ Requirements

 - ★ Challenges

 - ★ Example: vSphere

 - ★ **The Web view**

 - ★ Current situation

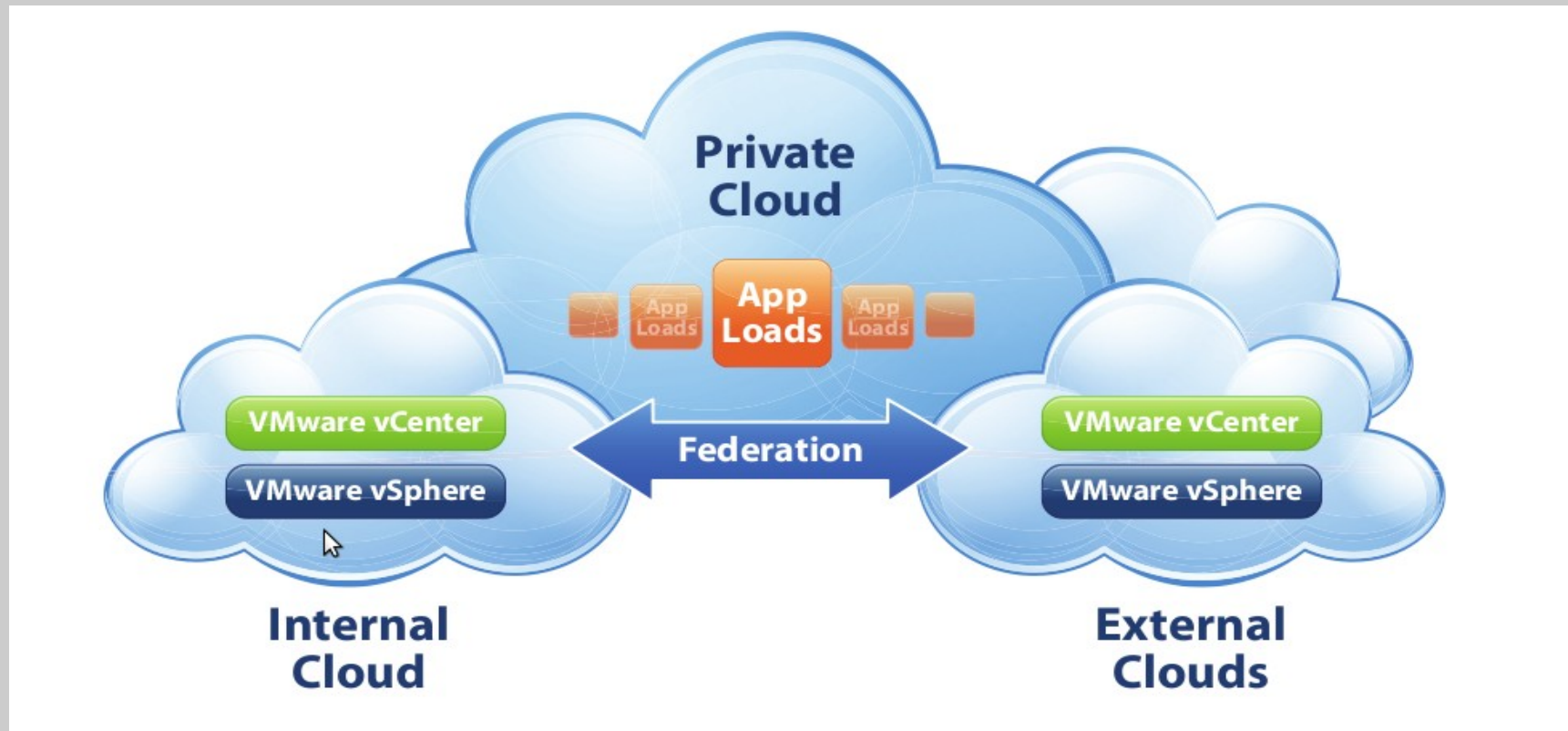
 - ★ Goals & Obstacles

 - ★ Example: Gazelle

- ★ **Conclusions**

An OS for the Datacenter

*“A **secure** computing environment that allows computing capacity from both internal and external clouds to **inter-operate** and be delivered much like a **utility**”*



Motivation (1)

- ★ The dark side of IT:
 - ★ **complex, hard-coded and inflexible.**
- ★ Existing public cloud solutions (Amazon EC2, Google Apps) require rewriting the applications.
- ★ Can businesses afford the **time, money** and **disruption**?
- ★ And if yes, how can we avoid **lock-in**?

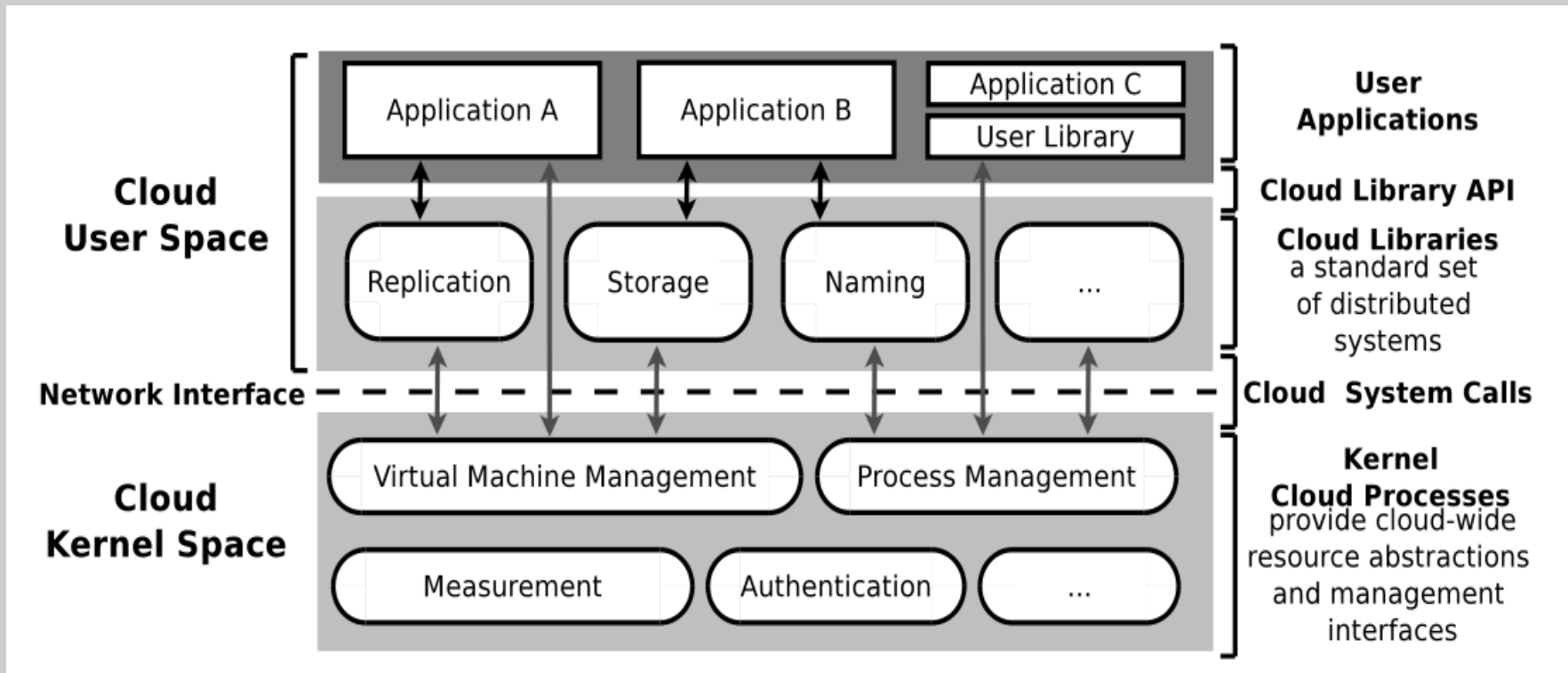
Motivation (2)

- ★ Enterprises need a cloud solution
 - ★ Non-disruptive
 - ★ Compatible with existing applications
 - ★ Enterprise-ready
 - ★ Allowing choices
 - ★ Avoiding lock-in

Requirements

- ★ Allocation/Deallocation of virtual machines
- ★ Process migration
- ★ Inter-process communication
- ★ Network-based interfaces to manage resources
- ★ Software support for the deployment of distributed applications

A sample architecture



- ★ Resource management using measurements and/or virtual coordinates
- ★ Process and VM management
- ★ Access control and user authentication

Challenges

- ★ Autonomous resource management
- ★ Node failure tolerance
- ★ Federation with external cloud in a secure way
- ★ Scalability

Example: VMware's vSphere

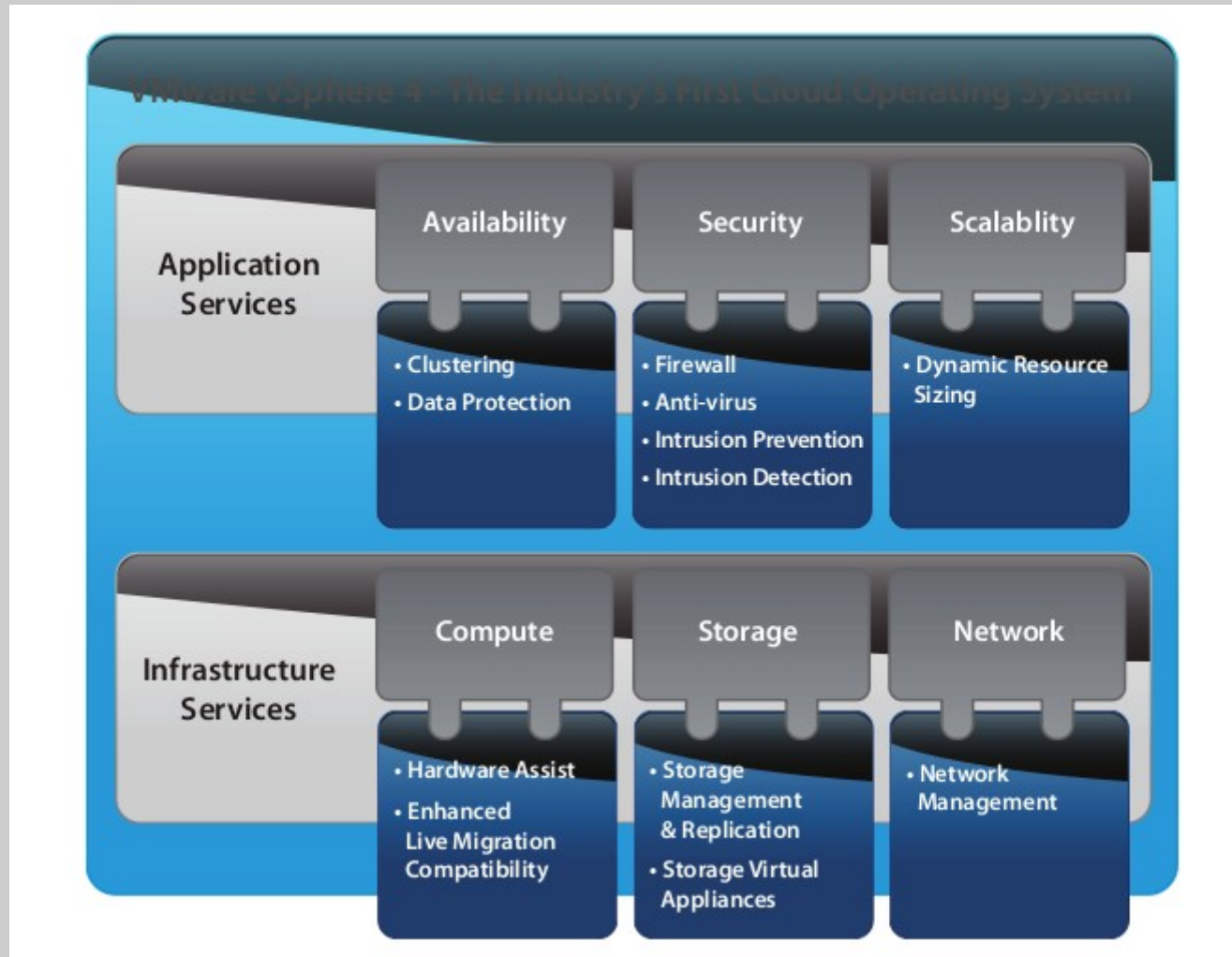
★ Goals:

- ★ Aggregate the infrastructure of an entire datacenter
- ★ Create a single and powerful “compute plant”
- ★ Quick and dynamic resource allocation

★ Services:

- ★ Infrastructure Services
- ★ Application Services

vSphere Architecture



vSphere: Infrastructure Services

- ★ Virtualize Server Resources
 - ★ Organize virtual servers into easily allocated pools
 - ★ Monitor power consumption to maximize energy efficiency
- ★ Virtualize Storage
 - ★ Abstract resources from underlying hardware
- ★ Virtualize Network
 - ★ Tools to optimize administration and management of virtual network environments

vSphere: Application Services

★ Availability

- ★ Priority scheduling
- ★ Eliminate data loss and unscheduled downtimes
- ★ Fault tolerance and data recovery mechanisms

★ Security

- ★ Enforce corporate security policies at the application level
- ★ Network segmentation of users and sensitive data

★ Scalability

- ★ Load-balancing
- ★ “hot add” and “hot plug”

Outline

- ★ Introduction

- ★ **Cloud Operating Systems**

 - ★ The Datacenter view

 - ★ Motivation

 - ★ Requirements

 - ★ Challenges

 - ★ Example: vSphere

 - ★ The Web view

 - ★ Current situation

 - ★ Goals & Obstacles

 - ★ Example: Gazelle

- ★ Conclusions

Outline

- ★ Introduction

- ★ **Cloud Operating Systems**

 - ★ The Datacenter view

 - ★ Motivation

 - ★ Requirements

 - ★ Challenges

 - ★ Example: vSphere

 - ★ **The Web view**

 - ★ Current situation

 - ★ Goals & Obstacles

 - ★ Example: Gazelle

- ★ Conclusions

Ready for a Web OS?

★ In 1995

- ★ Marc Andreessen predicted that Netscape would one day render Windows obsolete



★ Today

- ★ Desktop OS still prevails

A new generation of browsers

★ New Web Standards



- off-line support
- local storage
- geolocation capabilities
- graphics acceleration
- access to client devices

★ New Experimental Technologies

- web apps as native x86 code
- eliminate script-languages
- run an interpreter in the browser



★ Architectural Changes

★ Process-oriented browsers

Goals & Obstacles

- ★ Rapid deployment
- ★ Rich and interactive interfaces
- ★ Side-step compatibility issues
- ★ Security
- ★ Device integration
- ★ Performance

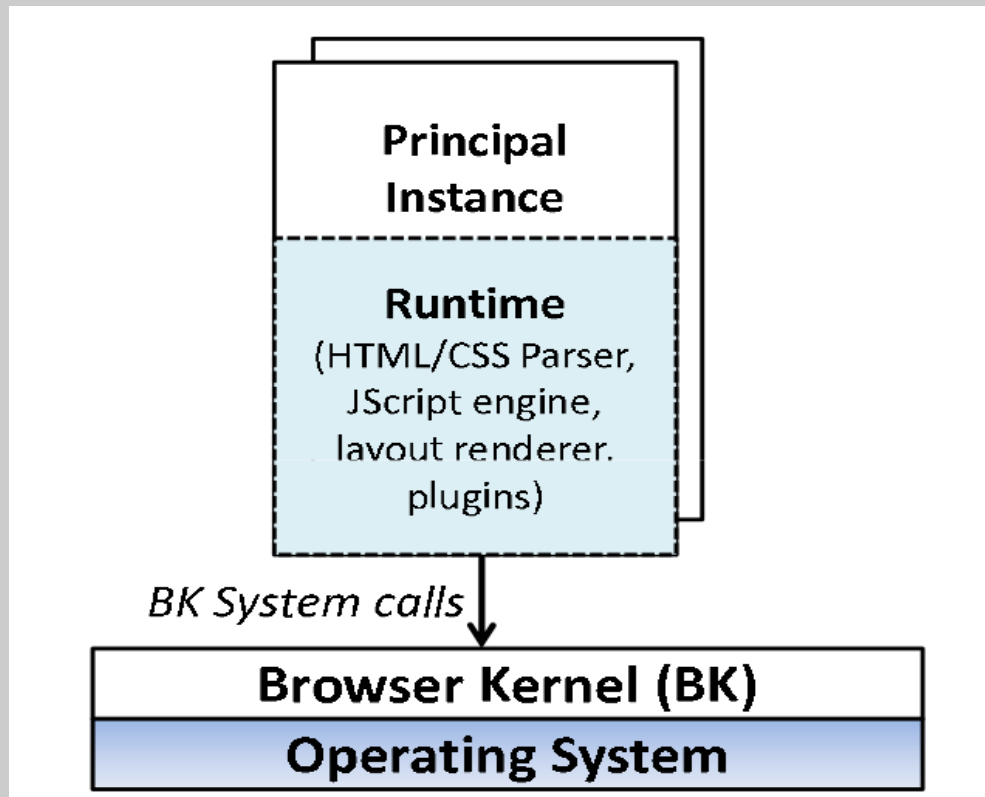
Example: Microsoft's Gazelle

- ★ Browser Kernel
 - ★ Separate protection domain
 - ★ Interacts directly with underlying OS
 - ★ Exposes a set of system calls
- ★ Processes communicate through the Kernel
- ★ Same Origin Policy principal
 - ★ `<protocol, domain_name, port>`

Microsoft's Gazelle: Process Models

- ★ Process-per-site
 - ★ *alice.profiles.socialnet.com* and *socialnet.com* share the same principal
 - ★ Embedded components, such as `iframes`, are placed in the same process despite their origin
 - ★ Plugins share a plugin process
 - ★ Used by Google Chrome
- ★ Process per tab
 - ★ Used by IE 8
- ★ Gazelle uses 2 processes-per-principal & SOP

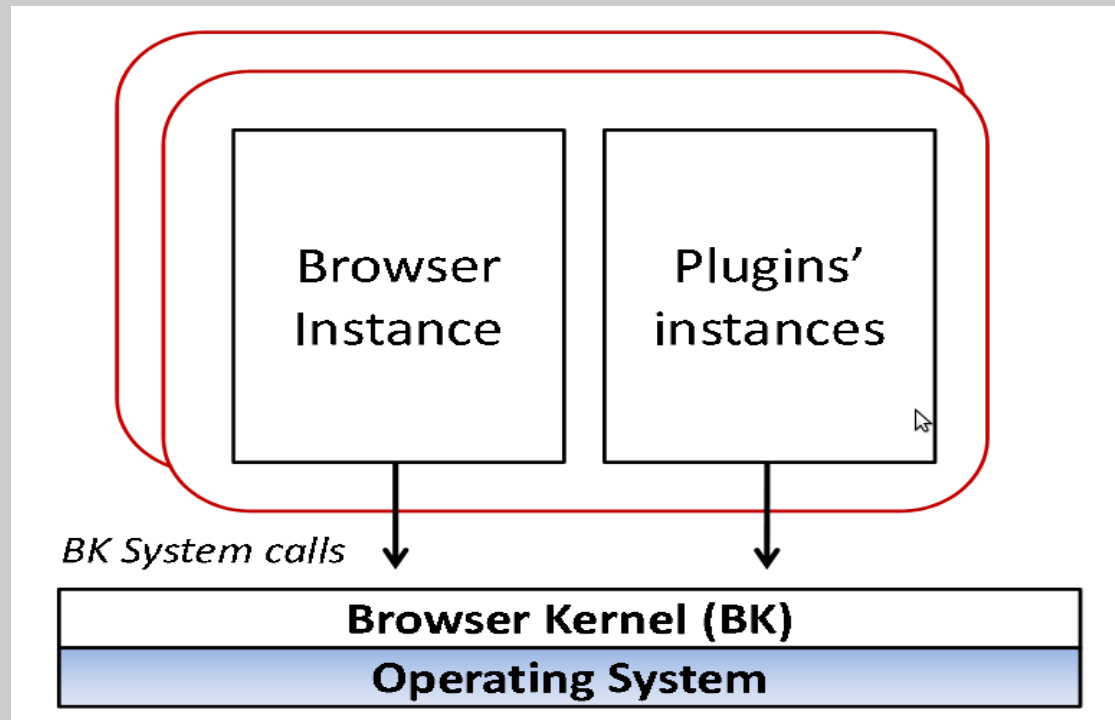
Microsoft's Gazelle: Simple Architecture



- ★ Principal is a unit of
 - ★ protection
 - ★ failure containment
 - ★ resource allocation
- ★ Message-passing communication

Microsoft's Gazelle: Supporting Legacy Protection

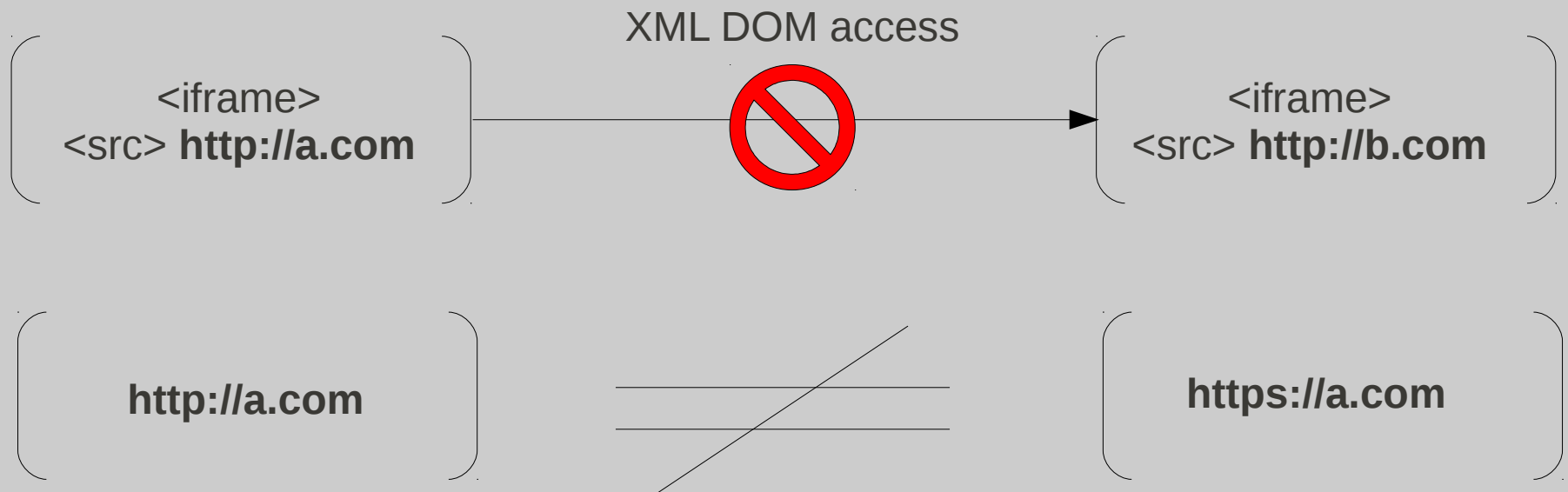
- ★ Numerous web sites rely on cross-origin scripts
- ★ Plugin instances can only interact with cross-origin scripts and style sheets through the browser instance



Microsoft's Gazelle: Security (1)

★ Scripts

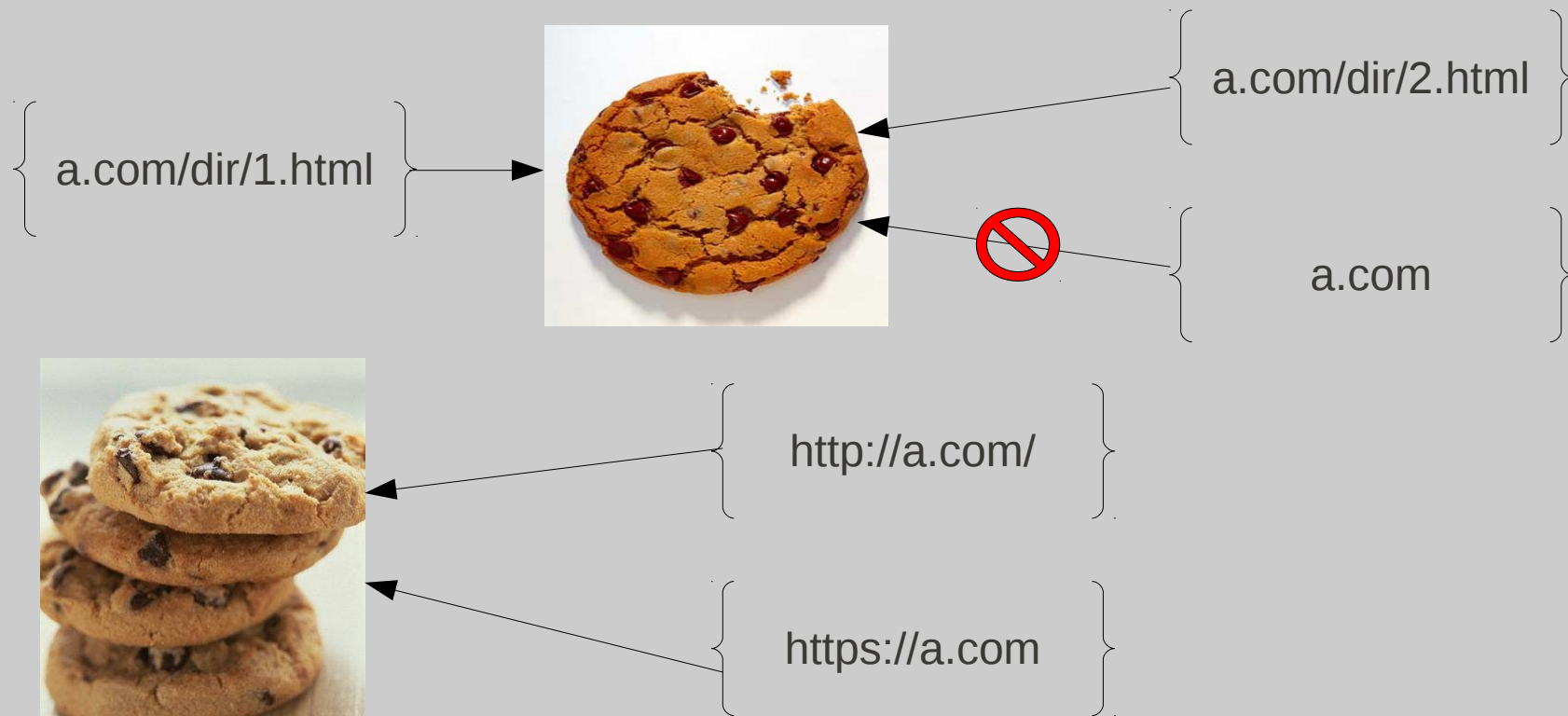
- ★ Two documents with different origins cannot access each other's HTML documents using DOM



Microsoft's Gazelle: Security (2)

★ Cookies

★ Host name and path



Microsoft's Gazelle: Performance

	Gazelle		IE 7		Google Chrome	
	Time	Memory	Time	Memory	Time	Memory
1. Browser startup	668 ms	9 MB	635 ms	14 MB	500 ms	25 MB
2. New tab (blank page)	602 ms	14 MB	115 ms	0.7 MB	230 ms	1.8 MB
3. New tab (google.com)	939 ms	16 MB	499 ms	1.4 MB	480 ms	7.6 MB
4. Navigate to google.com/ads	955 ms	6 MB	1139 ms	3.1 MB	1020 ms	1.4 MB
5. Navigate to nytimes.com	5773 ms	88 MB	3213 ms	53 MB	3520 ms	19.4 MB

2 process creations

Reuse of the previous tab process

Conclusions

- ★ Cloud Computing is here today.
- ★ Private Clouds can offer to the businesses a painless way to Cloud Computing but providers need to agree on federation policies and common management services.
- ★ Web OSes still have important challenges to face but new generation web browsers show that convergence is possible.

References

- [1]: VMware White Paper: “VMware vSphere, the First Cloud Operating System, Provides an Evolutionary, Non-disruptive Path to Cloud Computing”
- [2]: Fabio Pianese, Peter Bosch, Alessandro Dominuco, Nico Janssens, Thanos Stathopoulos, Moritz Steiner, “Toward a Cloud Operating System”
- [3]: Alex Wright, “Ready for a Web OS?”, Communications of the ACM, December 2009, Vol. 52, No. 12
- [4]: Helen J. Wang, Chris Grier, Alexander Moshchuk, Samuel T. King, Piali Choudhury, Herman Venter, “The Multi-Principal OS Construction of the Gazelle Web Browser”