High Availability CAS

Adam Rybicki, Scott Battaglia

2009 Jasig Conference, Dallas, TX

March 4, 2009

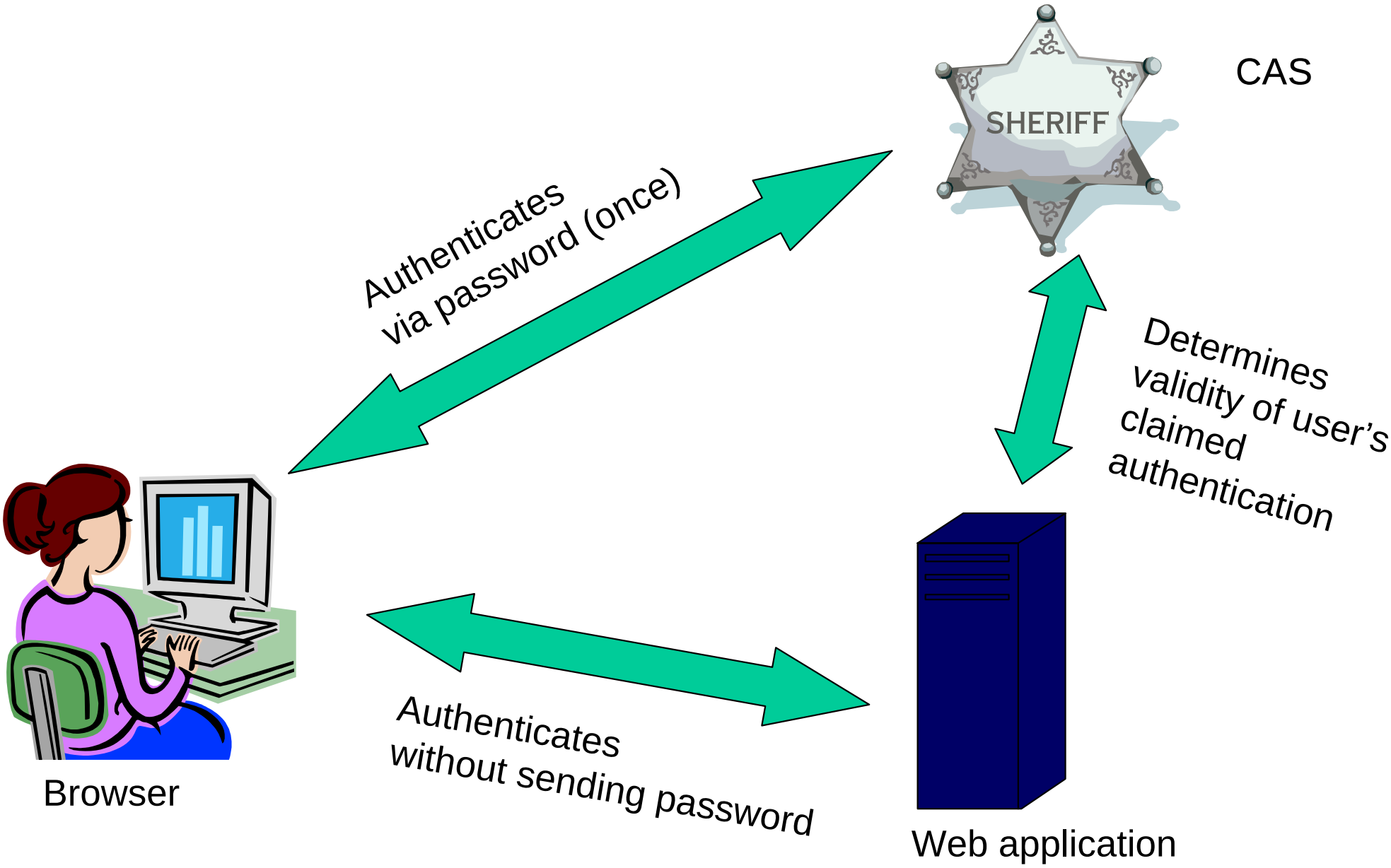
UNICON

1. Overview
    a. Introductions
    b. Definitions
2. Part 1: Clustering CAS (uptime)
    a) Methods From CAS Manual
    b) Other Tried Options
3. Part 2: High Availability (now that it stays up)
4. Conclusions

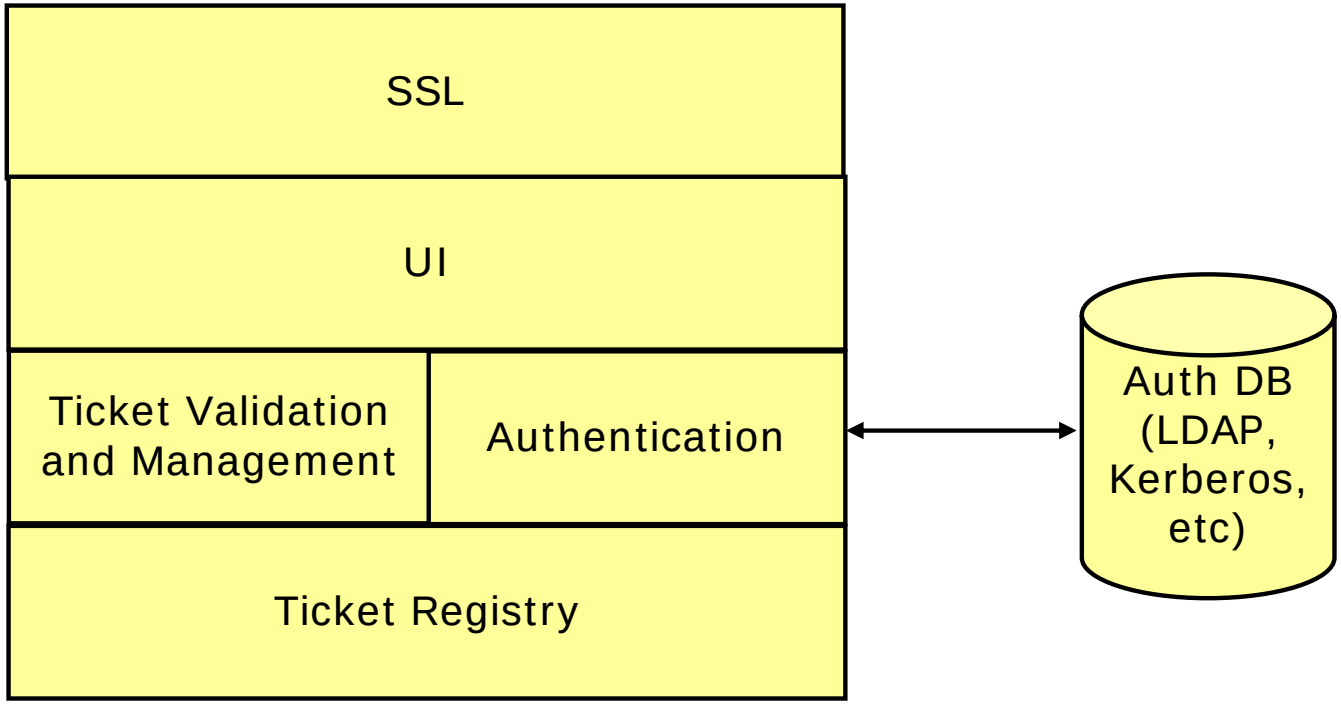
Overview

- CAS Overview
  - Challenges unique to clustering CAS
  - How is CAS availability handled today?
- What Is High Availability?
- High Availability vs. Uptime
- CAS Manual Addresses Only Uptime

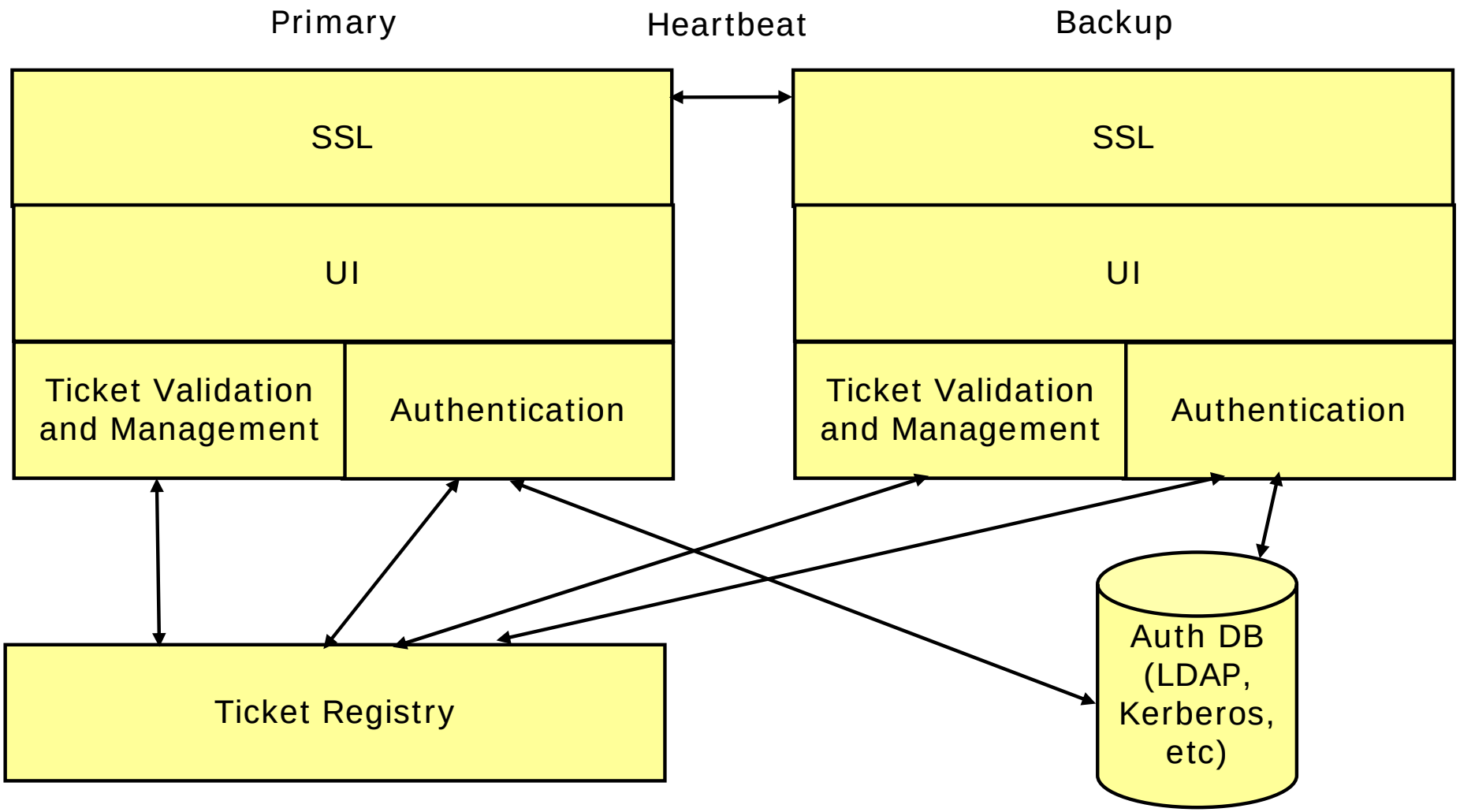
CAS in a nutshell



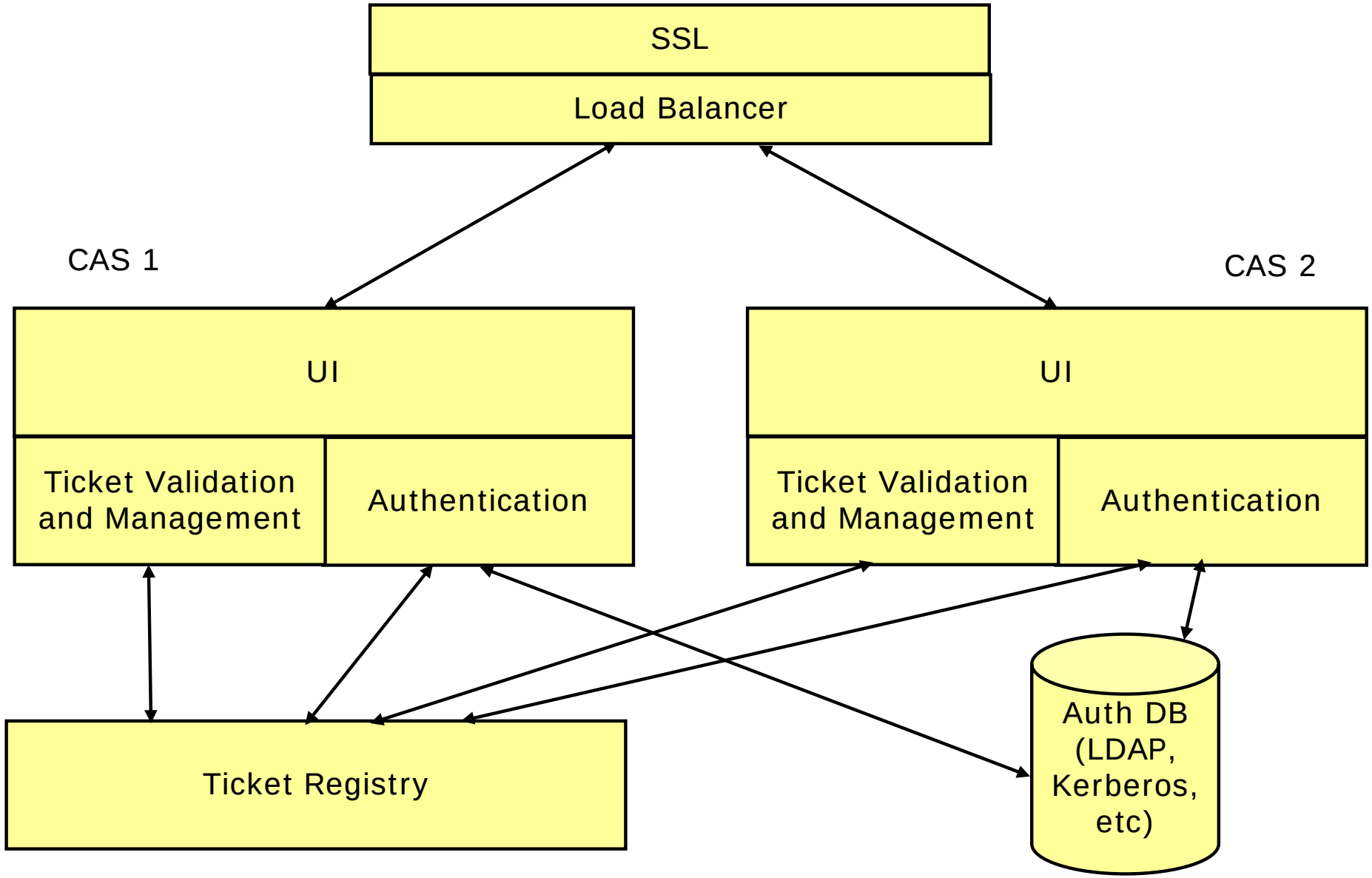
Single CAS Server



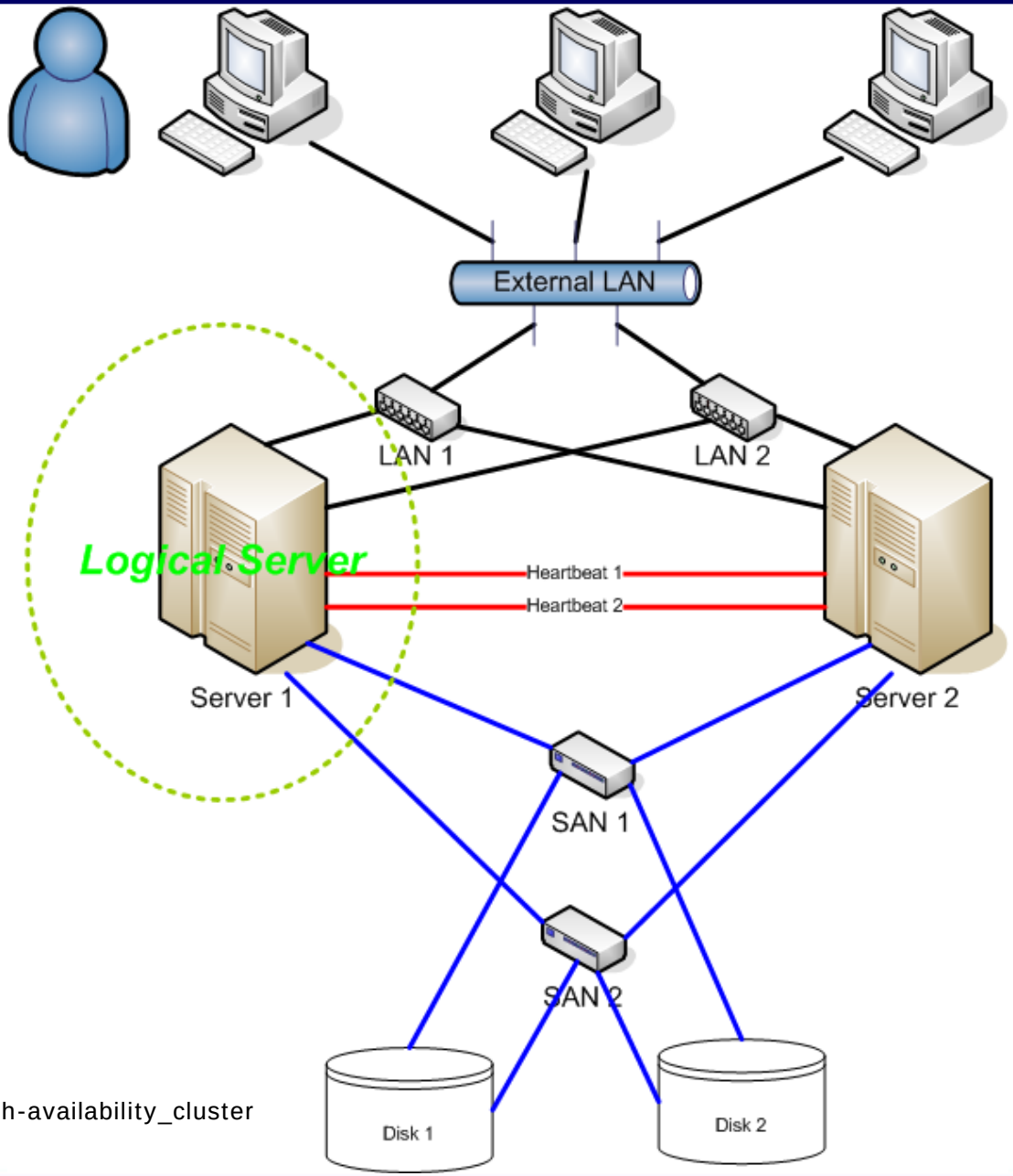
Primary with a Backup



Load-Balanced Cluster



High Availability



http://en.wikipedia.org/wiki/High-availability_cluster

Part 1: Clustering CAS According to the CAS Manual

- Guarantee ticket uniqueness
  - set host.name in cas.properties
- Distributed ticket registry
- Tomcat session replication
  - Why this is not really necessary
    - It complicates things considerably
    - It can be eliminated with this simple change:

```
in WEB-INF/cas-servlet.xml set
<flow:executor id="flowExecutor" registry-ref="flowRegistry" repository-type="client">
```

Distributed Ticket Registry

- CAS Manual Only Talks About JBoss Cache Ticket Registry
- There are at least 2 other ***tested*** options:
  - JPA Ticket Registry
  - Memcache Ticket Registry

JBoss Cache Ticket Registry

- Problems reported using the default JBoss Cache 1.4.1 (prior to CAS 3.3 release)
- Unlimited number of nodes
- Default version does not persist the cache
- Default Registry Cleaner may be OK, and it may be OK to run it on all cluster nodes

Memcache Ticket Registry

- Memcached is a high-performance distributed memory object caching system
  - Memcached assumes that objects may be reloaded from persistent storage when not in cache
  - Scales by adding more nodes
  - Replication possible with the repcached patch
  - If it runs out of memory, it throws away LRU objects
  - Many client libraries, including one in Java
- Impossible to run a registry cleaner
- Memcache has its own expiration/cleaner

JPA Ticket Registry

- Surprisingly good throughput shown in testing under moderate load (used MySQL in that test)
- Concurrent access neatly solved using database transactions and Hibernate
- Default registry cleaner insufficient
- Now all you have to do is to make your database HA

Other Possible Methods

- Terracotta (http://www.terracotta.org/)
  - Sample configuration contributed (http://www.ja-sig.org/wiki/x/ihjP)
  - Requires the use of Terracotta server(s)
- Ehcache (http://ehcache.sourceforge.net/)
  - Just like JBoss Cache, does not require dedicated servers
  - If uPortal can use it, why not CAS?
- Apache Tribes (http://tomcat.apache.org/tomcat-6.0-doc/tribes/introduction.html)
  - That's the only publically-accessible documentation
  - No dedicated server required

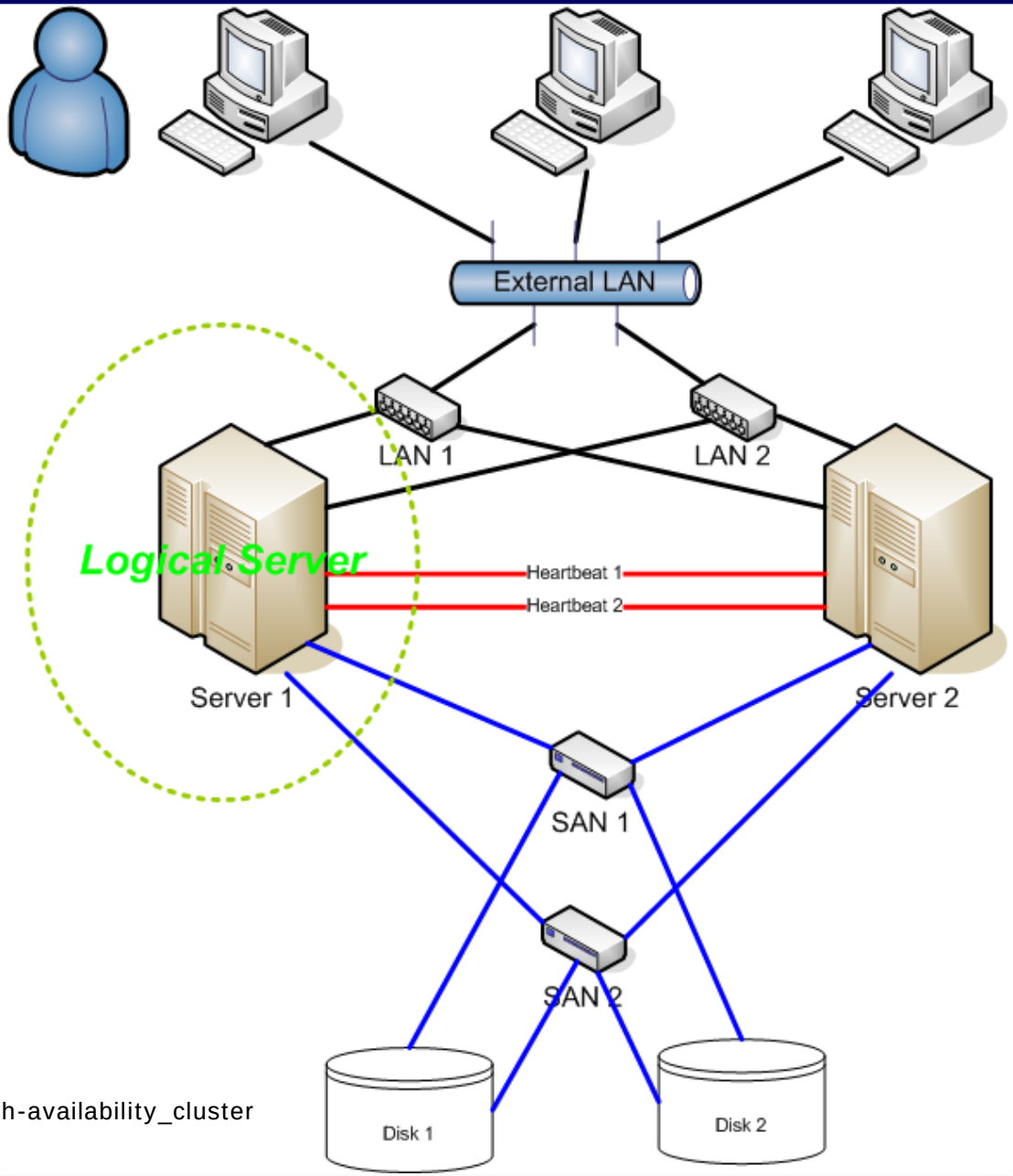
Registry Cleaner Limitations

- Running a cleaner on two cluster nodes may not work
- The current TicketRegistry permits the retrieval of all tickets
- Memcache does not allow to retrieve all entries
- JBoss registry allows it, but it may take too long to process
- Default cleaner can stall database access because it goes through the entire registry in 1 transaction

Part 2: Items Not Addressed by the CAS Manual

- Load-balancing
- Fail-Over of CAS
- Single Sign-out
- Authentication Database Availability
- Network
- Power

High Availability



http://en.wikipedia.org/wiki/High-availability_cluster

Load-balancing

- Should Be Easy to Solve With Apache Using AJP Protocol
- Allows to offload SSL server processing to Apache
- mod_proxy Now Supports AJP
- Simple HTTP Proxy Not Recommended
- mod_jk May Still Be More Flexible than mod_proxy

Fail-over of CAS

- Is Apache a Single Point of Failure?
- Can Use Linux-HA to Watch the Apache Server
  - Have a "Spare" or "Standby" Server Available
  - Linux-HA Does IP Swap of a Failed Server
  - After The Swap the Standby Activates
- FOSS!

Authentication Source

- LDAP (using OpenLDAP)
  1. Can be dealt with using Linux-HA
  2. Can configure CAS to authenticate to several LDAP servers
- Kerberos
  - JAAS can be configured to authenticate to several KDCs
- Database
  - Database-specific ☺

Network

- Use Multiple Locations
- Use Multiple Interfaces
- Use Multiple DNS Records
- Use Multiple ISPs

Power

- UPS
- Generators
- Multiple Locations
- Multiple Power Suppliers

Conclusions

- A Complete CAS Cluster Can Be Implemented With FOSS
  - Yes, there is investment in expertise
  - Proprietary software expertise is not free
  - Commercial support for FOSS is available
- Do Not Try to Replace Proven Commercial Software With FOSS Just Because It's Free
- Try to Follow Methods Proven at Your Institution

Questions?

Adam Rybicki

arybicki@unicon.net

www.unicon.net

UNICON®