What went wrong?

Error handling in a distributed environment

OT2004 Workshop

Andy Longshaw and Eoin Woods

The Premise

- Timely and appropriate reporting of error conditions is vital for any significant application
- One component/application on one machine presents few problems
- Distributed applications consisting of many components (some 3rd party) are a different matter

- What are the problems and how can we address them?

The Objectives

- Identify good/best practices for identification and management of errors in a distributed environment
- Identify bad practices (what doesn’t work) so it can be avoided
- See if any coherent framework or set of proto-patterns presents itself

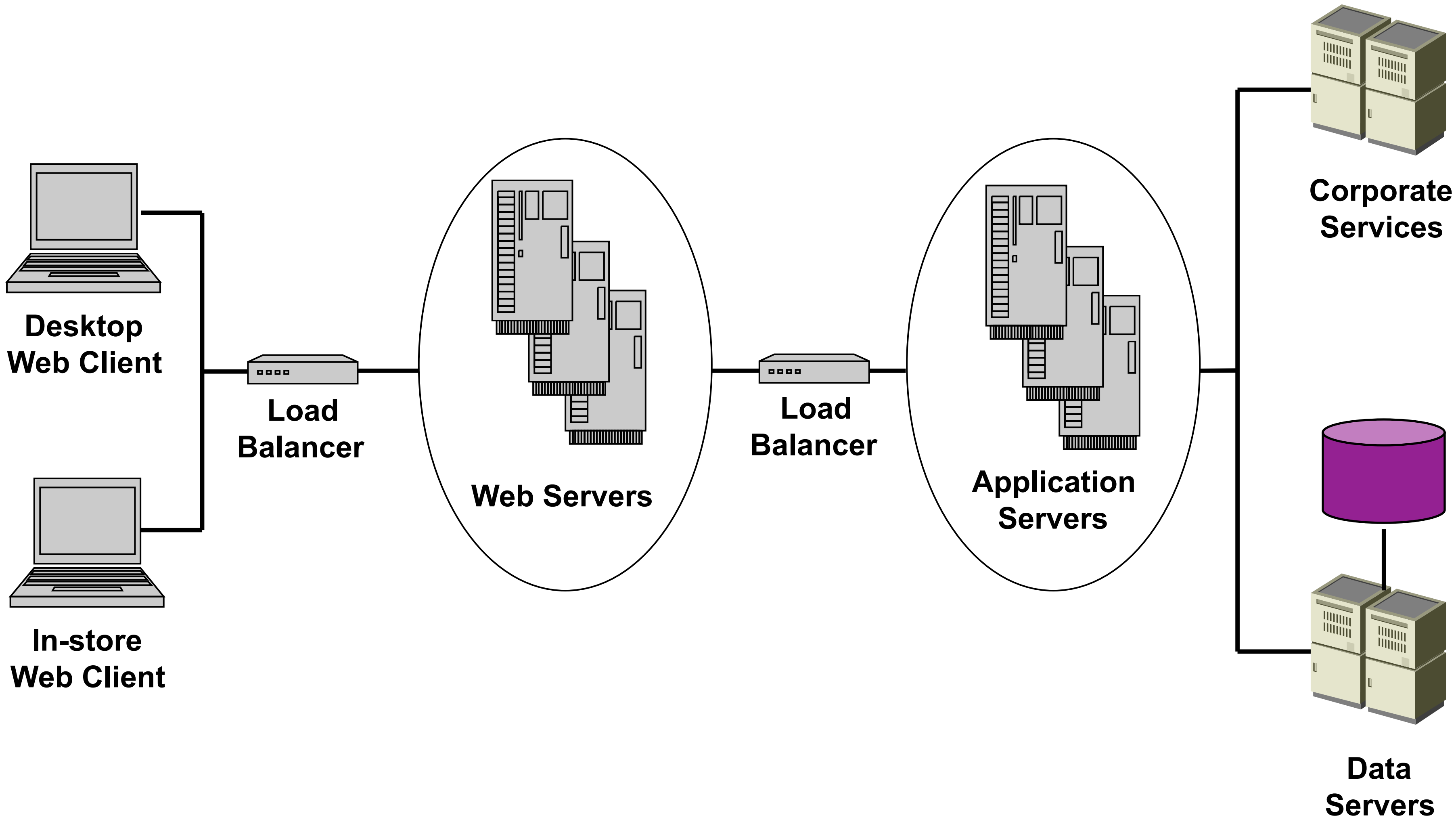
How do we do this?

A&E	0:00-0:10	Introduction and objectives
A	0:10-0:25	Case study in distributed error handling
G	0:25-0:55	“The rough with the smooth”
*	0:55-1:15	Collate and discuss list of practices
*	1:15-1:45	Break
G	1:45-2:20	Framework design
*	2:20-2:40	Presentations (5 mins per-group)
*	2:40-3:00	Closing discussion and conclusions

Case Study

- Example of distributed development
- Challenges encountered in error handling and problem identification
- Some suggested solutions

Case Study: The scenario



Case Study: General issues

- Local reporting vs. distributed
  - Network issues and contacting central servers
  - Collation of distributed logs
- Different logging mechanisms
  - Web logs
  - System error logs
- Time consistency between machines
- Web services and error propagation
- Exceptions due to distribution
- Error levels and importance

Case Study: Project-specific issues

- To pass exceptions or not to pass exceptions
  - What is the meaning of exception to another tier?
  - Pain of passing across Web Services
  - Where and when to log
- ‘Functional’ exceptions vs. ‘system’ exceptions
- Independence of services vs. all-encompassing error handling framework
- Are all exceptions exceptional?
- Tracing errors across multiple tiers of load-balancing
- Where did real error occur (client, web, app, etc.)?
- Integration with error mechanisms of external systems

Case Study: Stakeholder viewpoints

- Application support team (3rd line)
  - As much information as possible to debug problem
  - Consistent mechanism to copy for new code
- Infrastructure support team
  - Must plug into monitoring tool (OpenView)
- Business sponsor
  - No impact on performance while maintaining ‘fixability’
- End users
  - Just fix the problem

Case Study: Requirements ‘next time’

- Consistent mechanism for reporting errors
- Concise information with ability to drill down if needed
- Fixed rules for developers about when and where to log errors
- Creating a single ‘System Overview’ including multiple elements and external systems
- Consistent user experience

Example of what we did (1)

- Log technical errors at distribution boundary
- What we did
  - Log error information in Remote Façade
  - Pass back error code
- Motivation
  - Consistent location for error logging
  - Detail stays on machine where error occurred

Example of what we did (2)

- Log propagated call id
- What we did
  - Generate a unique call ID when client request hits presentation layer
  - Propagate call id across all tiers
  - Log call id when errors occur
- Motivation
  - Associate logged errors together
  - Trace problematic call across multiple tiers
  - Detect patterns and “knock-on” errors

Ex1: The Rough with the Smooth

- ***Aim***: identify good and bad practices for error handling
- ***How***: in groups, discuss the error handling approaches you've used
  - What worked?
  - What didn't work?
- ***Output***: a list of good and bad practices, with reasons

Good and Bad Practices

Group Discussion

Ex2: Framework Design

- ***Aim***: design an error handling framework!
- ***How***: in groups:
  - Summarise the good/bad practices to requirements
  - Add another non-functional requirement
  - Sketch the design of a framework to solve this problem
- ***Output***: a memorable presentation of your framework to the entire group

Presentations

5 minutes per group

Discussion and Conclusions

What have we learned?

What general lessons can we take away?