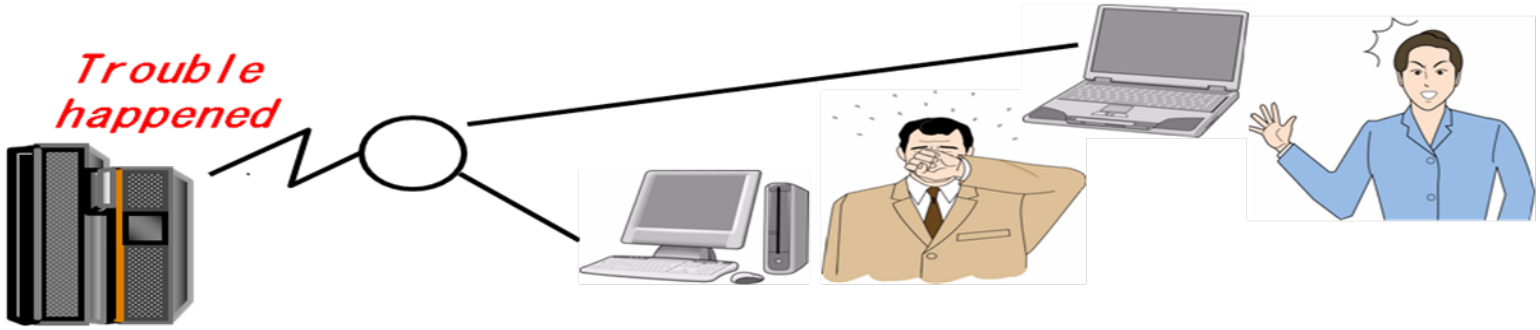
IT Architecture

Component Failure Impact Analysis (CFIA)

Dr. Marcel Schlatter,
IBM Distinguished Engineer
Member of the IBM Academy of Technology
marcel.schlatter@ch.ibm.com

The Problem



- One single component failure has brought the whole system down
- To repair one single component, the whole system had to be stopped.
- End-users suffer from long system/network down times.
- System and network documentation not up-to-date, does not show changes made since system was originally designed and built.
- No, or no up-to-date recovery procedure manual is available
- Following a component failure, only a specific person can recover the system

Component Failure Impact Analysis (CFIA)

- A **proactive** Availability Management method
  - Provide IT with the business and user perspective about how deficiencies in the infrastructure and underpinning process, procedures, and service delivery skills impact the business operation.
  - The use of business-driven metrics can demonstrate this impact in real terms and help quantify the benefits of improvement opportunities.
- A method to optimize the capability of the IT infrastructure, services and supporting organization to deliver a cost-effective and sustained level of availability enabling the business to meet their objectives
- A risk assessment methodology that provides a systematic way to thoroughly review failure modes of complex interacting system components, and the effects of failures on the overall system.

Purpose of a CFIA, and how the results from a CFIA can be used

The purpose of a CFIA is to provide information to ensure that the availability and recovery design criteria for new or existing IT services are influenced to **prevent** or **minimize the impact** of failure to the business operation and users.

- CFIA can be used to predict and evaluate the impact on IT service arising from component failures within the technology.
- The output from a CFIA can be used to identify where additional **resilience** should be considered to **prevent** or **minimize the impact** of component failure to the business operation and users, and where enhancements should be made to **procedures** or **processes** and **skills**.
- This is particularly important during the Design stage, where it is necessary to predict and evaluate the impact on IT service availability arising from component failures within the proposed IT Service Design.
- However, the technique can also be applied to existing services and infrastructure, as a way to measure the technical and service delivery health of an existing environment.

Major goals, and typical results when a CFIA is used to measure the technical and service delivery health of an **existing environment**

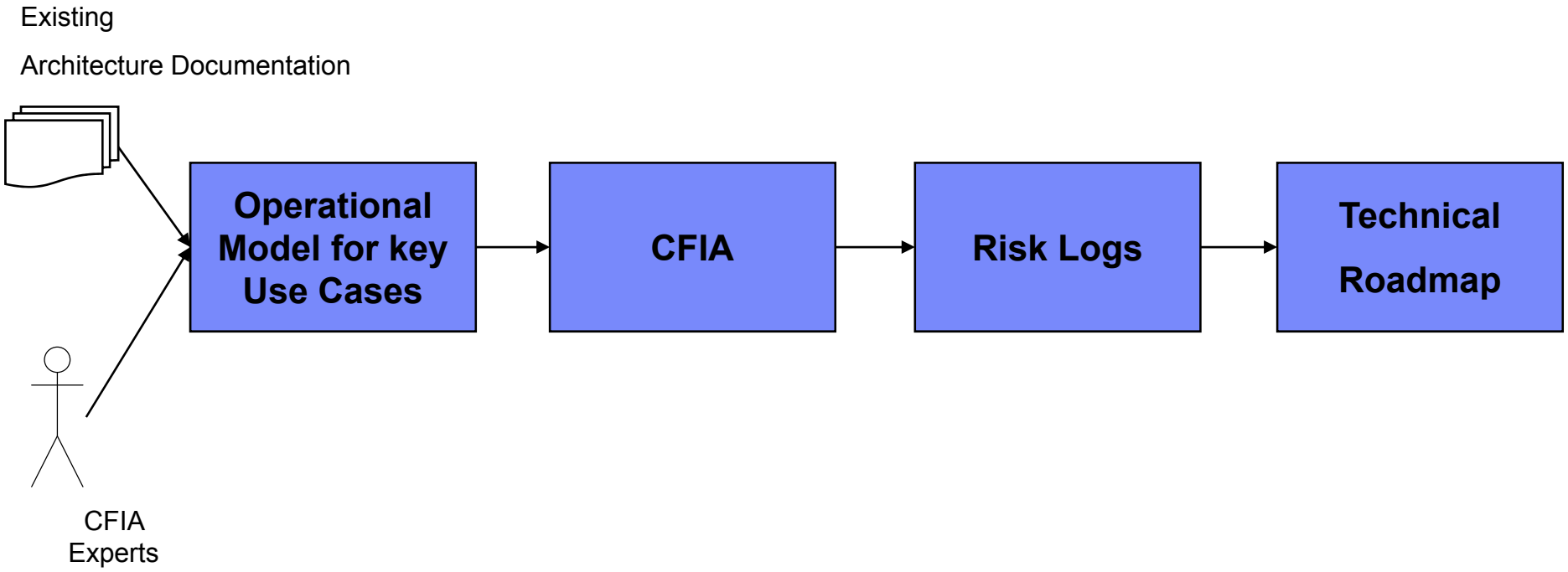
Major goals of a CFIA

- Identify critical availability and performance **risks** for some or all business critical applications
- Identify measures that can be taken to **reduce the probability of failures**
- Identify measures that can be taken to **reduce the time it takes to detect and repair** those failures

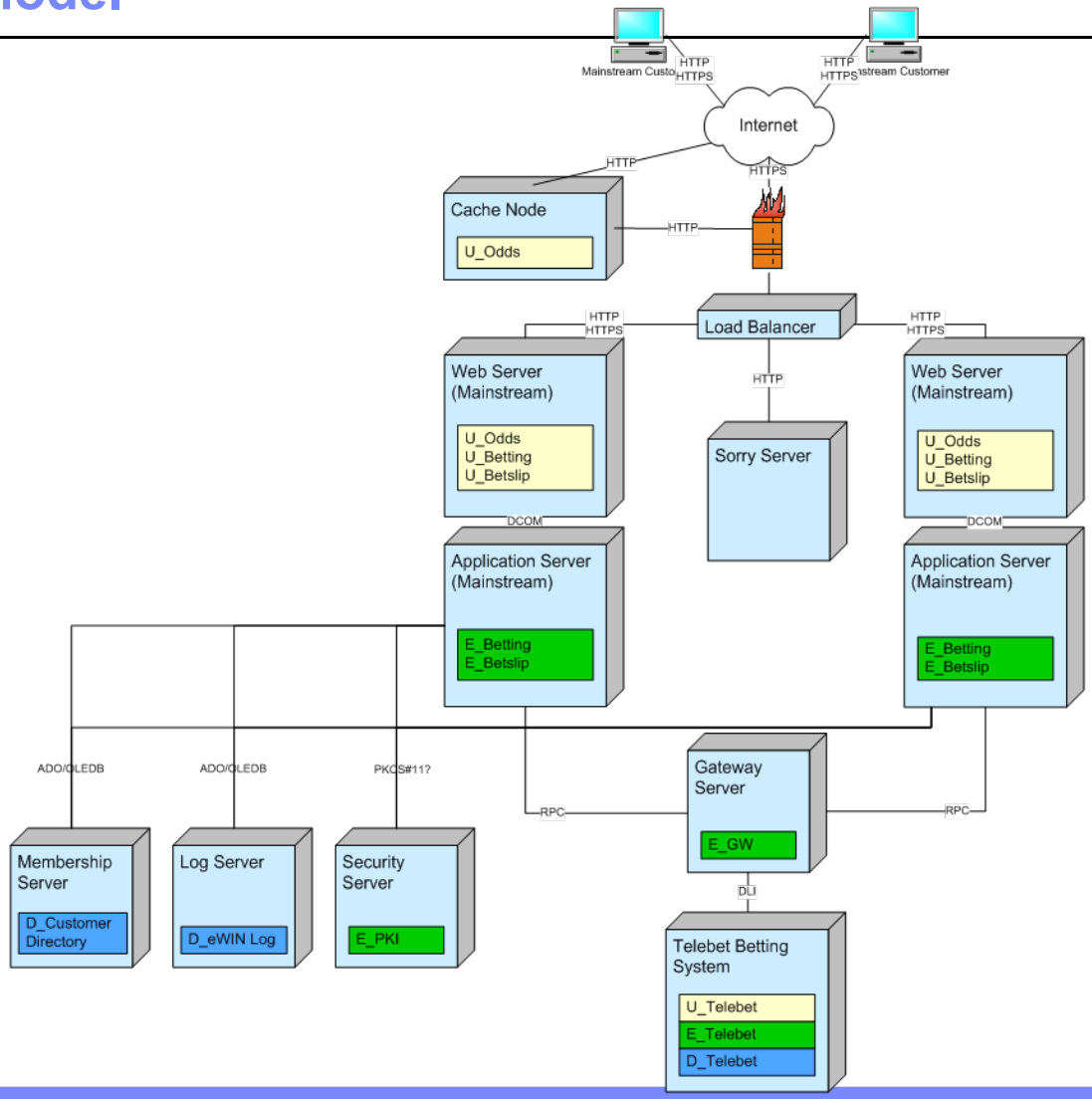
Typical result: a phased solution roadmap that covers technical, organizational and process aspects

- **Phase 1:** Improvements that must and can be achieved immediately, e.g., within a few weeks.
- **Phase 2:** High-priority improvements that must and can be achieved in the medium term, e.g., within a few months.
- **Phase 3:** Lower-priority improvements, and improvements that require more fundamental architectural changes.

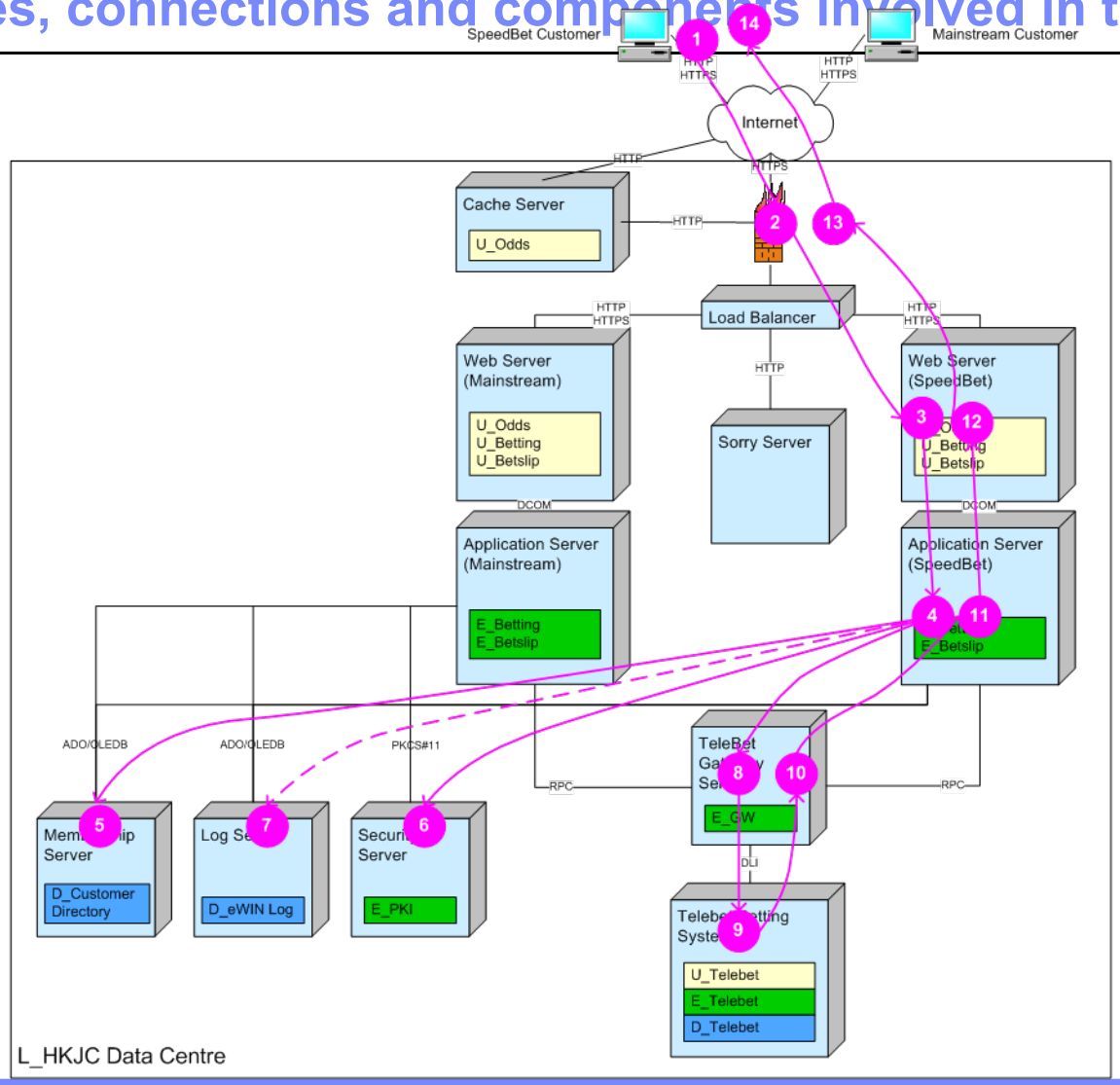
High level steps and deliverables



Example: Starting with a static view of the Logical/Specification Operational Model

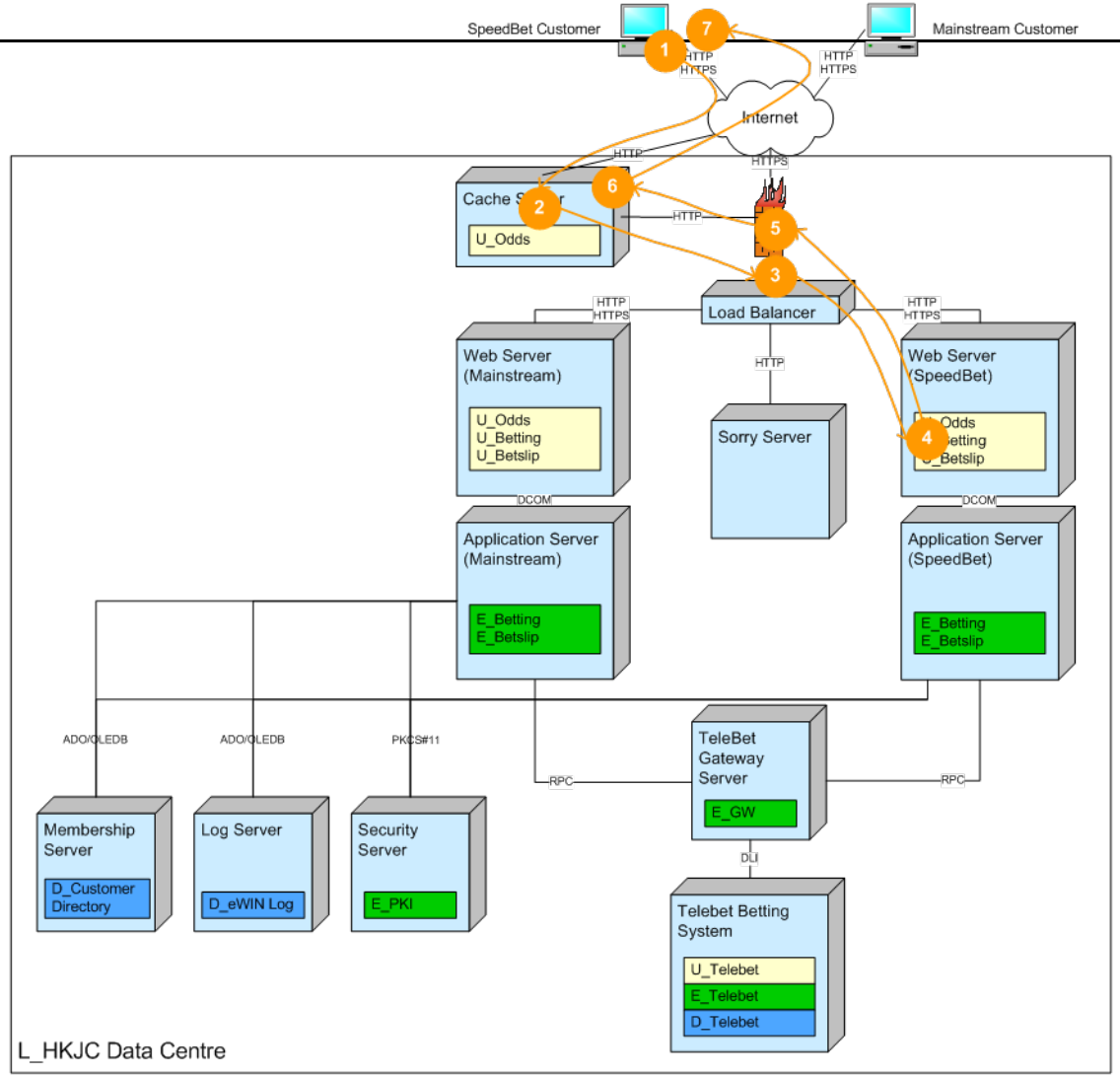


Walkthroughs of "Architecturally Significant" Use Cases to identify all the nodes, connections and components involved in the system



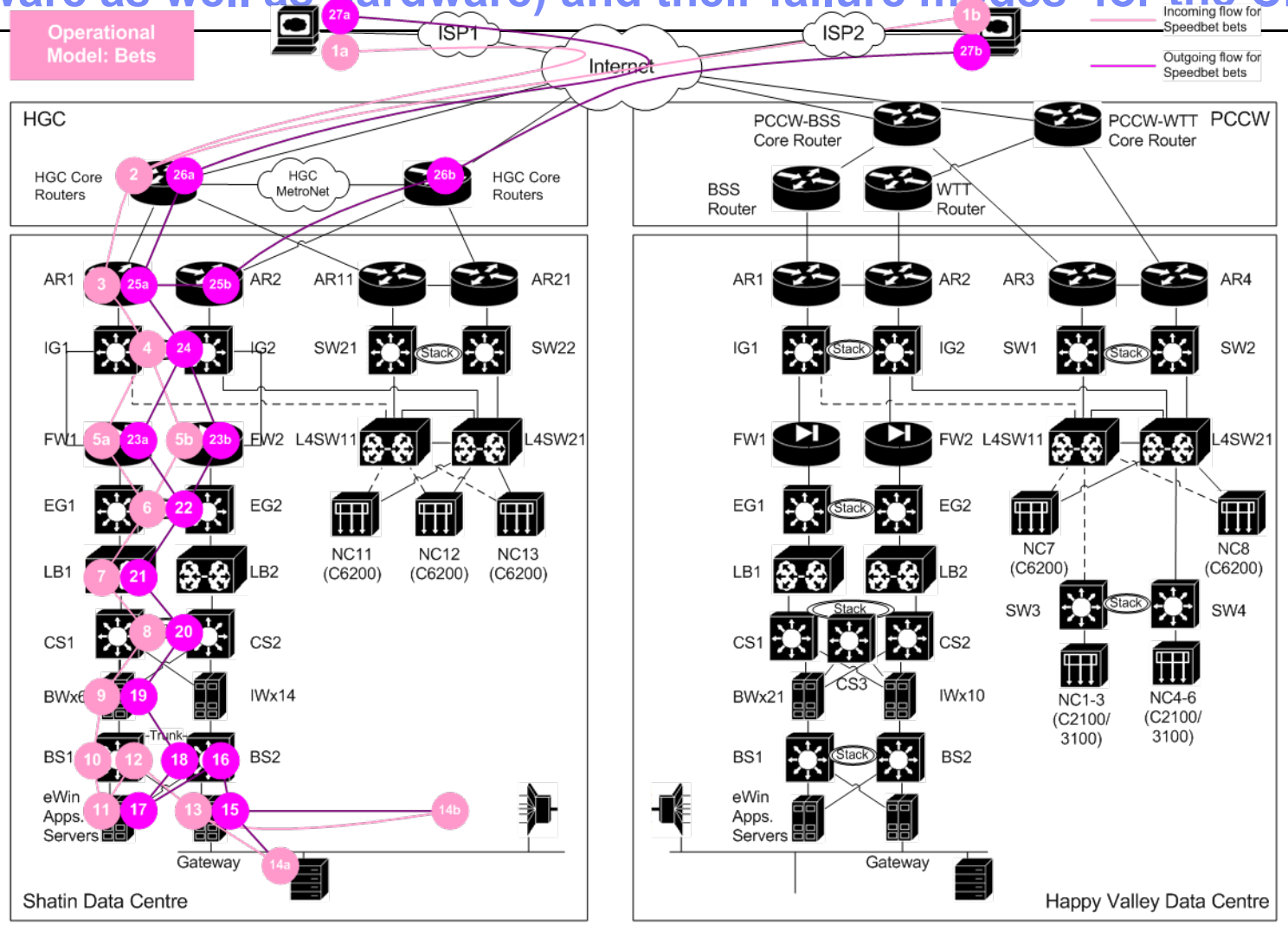
UC: Place Bet

In this case “architecturally significant” use cases are the minimum set of use cases needed to exercise all the components in the system



UC: Get Odds

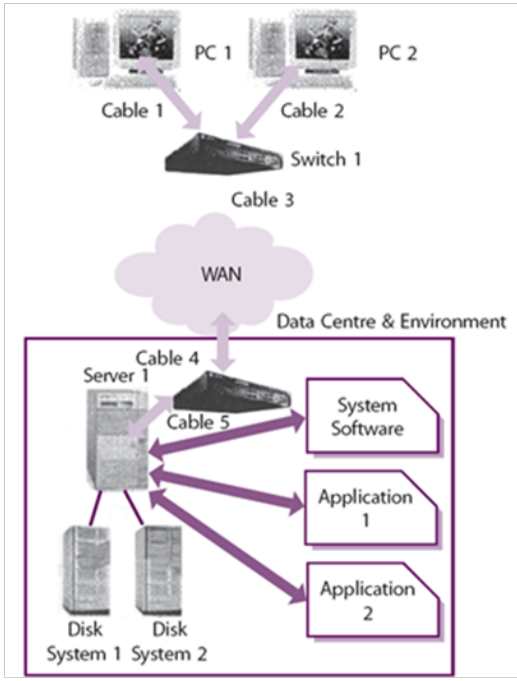
Elaboration to the physical level will identify all components (software as well as hardware) and their failure modes for the CFIA



UC: Place Bet

Basic CFIA Matrix

- Create a grid with Configuration Items (CIs, which can be components, nodes, links, etc., depending on the circumstances) on one axis and the IT services that have a dependency on the CI on the other.
- Populate the grid as follows:
  - Leave blank when a failure of the CI does not impact the service in any way
  - Insert an ‘X’ when the failure of the CI causes the IT service to be inoperative
  - Insert an ‘A’ when there is an alternative CI to provide the service
  - Insert an ‘M’ when there is an alternative CI, but the service requires manual intervention to be recovered.



CI	Service 1	Service 2
PC1	M	M
PC2	M	M
Cable 1	M	M
Cable 2	M	M
Switch 1	X	X
Cable 3	X	X
WAN	X	X
Cable 4	X	X
Switch 2	X	X
Cable 5	X	X
Data Centre	X	X
Server 1	X	X
Disk 1	A	A
Disk 2	A	A
System S/W	X	X
Application 1	X	
Application 2		X

CIs that have a large number of Xs are critical to many services and can result in high impact should the component fail.

IT services having high counts of Xs are complex and are vulnerable to failure.

Examples of Configuration Items (CIs)

- Hardware components such as servers, network cards, SAN switches, network switches, routers etc
- Operating system components such as the operating system itself and essential system services such as TCP/IP
- Middleware software components such as databases (eg. DB2, Oracle, SQL Server), application servers (eg. Weblogic, WebSphere, Tuxedo), message queues (e.g. MSMQ, MQ Series) and workflow managers (eg. WebSphere Process Server and Staffware)
- Application software such as SAP, Oracle Financials, Peoplesoft.
- Bespoke application components

Configuration Item (CI) (ITILv3): Any Component that needs to be managed in order to deliver an IT Service. Information about each CI is recorded in a Configuration Record within the Configuration Management System and is maintained throughout its Lifecycle by Configuration Management. CIs are under the control of Change Management. CIs typically include IT Services, hardware, software, buildings, people and formal documentation such as Process documentation and SLAs.

How to read the CFIA matrix

- Components that have a large number of Xs are critical to many services and can result in high impact should the component fail.

- IT services having high counts of Xs are complex and are vulnerable to failure.

This basic approach to CFIA can provide valuable information that helps to quickly identifying **SPoFs**, IT services at risk from CI failure and what alternatives are available should CIs fail.

It should also be used to assess the existence and validity of recovery procedures for the selected CIs.

Expand the CFIA Matrix with Failure Modes, Failure Effects, and Recovery Procedures

- For each component, describe:
  - What is it
  - How can it fail (Failure Mode)
  - How can component failure be detected (how fast, by whom, etc.)
  - What are the effects of the component failing
  - What are the recovery or failover procedures
- Ask yourself:
  - How do we respond when this CI fails?
  - What procedures do we follow? Are these procedures documented? Could they be improved? Could they be automated?
  - Can we improve the procedure through staff training? New tools or techniques?
  - Could preventative maintenance have helped avoid this problem?

An example of an expanded CFIA matrix

Logical Node|Component	Physical Component Name	Location	Zone	Component Description	Failure Mode	Failure Effects	What is the Recovery/Failover Procedure ?
MTN Active Reverse Proxy	dmzweb1	Newlands	DMZ	MTN Active landing page - active.mtn.co.za, HPBlade BL20p	Hardware failure IIS failure Performance degradation	All services in MTN Active become unavailable All services in MTN Active become unavailable Response times become degraded or MTN Active becomes unusable	No failover Recovery at DR site Manual problem diagnosis
MTN Active Servers	nldwlm01	Newlands	DC	Windows Server 2003, Enterprise Edition 5.2, SP1	Hardware failure	Uncertain - concerns over whether Weblogic 8.1.4 supports will reroute work if one of the servers fails	Restart Weblogic
	nldwlm02	Newlands	DC	Windows Server 2003, Enterprise Edition 5.2, SP1			

For all the components, he CFIA describes key impacts in response to different types of failures …

… and what the Recovery / Failure process is.

Node Analysis

- Failure types for the node.
- Impact of the node failing.
- Who is involved in the recovery of various components.
- Is there component-level redundancy.
- Are there component specific recovery processes.
- Are there node-specific failover or recovery procedures.
- Are there aspects of the nodes function that are prone to failure (e.g database log full etc).

Remember

In the **Operational Model**, a Node is defined as an aggregation of components and the hardware required to support them.

Components can be executables, or data.

Depending on the circumstances and the scope of the CFIA, investigate and document the following (1 of 2)

- **Node Description**
  - Node Name / Node ID
  - Primary Function of the Node
  - Business impact if node not available
  - Node Owner
  - Service Manager
  - Technical Lead
  - Failover Node(s) and how does it failover
  - Failover documents available: Y/N
  - Quality of the failover documents: Strong / weak
  - Date failover was last tested
  - Recovery method for H/W, S/W, and data
  - Recovery documents available: Y/N
  - Quality of the recovery documents (rate 1-10)
  - Data recovery was last tested
- **Hardware**
  - Hardware model and age in years
  - Rack or carcass
  - Processor: Single / SMP
  - Hardware Monitoring: Y/N; if yes: how
  - Heat Management: Y/N; Multiple Fans? Last checked?
  - Power Supply: Single / Dual
  - Internal Disk: Raid Level
  - Network card: Single / Dual
- **Operating System**
  - Version and Release
  - Supported: Y/N
  - Recovery Method: Image copy or re-install
  - Operating system recovery procedures, and roles and responsibilities: Strong / Weak
  - Patch Management: Strong / Weak
  - Privileged User ID Management: Strong / Weak

Depending on the circumstances and the scope of the CFIA, investigate and document the following (2 of 2)

- **Clustering**
  - Date last verified (e.g., parameter settings)
- **Applications and Data**
  - Backup method for failover / recovery: Tape, remote disk copy, sync/async, etc.
  - Failover Method
  - Failover tested? How? How long did it take?
  - Restore tested? How? How long did it take?
  - Data Loss **SLA?**
  - Application monitoring: How? What is monitored?
  - Data monitoring: How? What is monitored?
- **Operations and Admin Processes**
  - Strong / weak?
  - Experienced Operators
  - Experienced System Administrators
  - **RCAs** related to this node, or applications running on this node

Risk Log: Document key findings for each CI (Node, Component, Link, etc.) to support prioritization of risks, and selection of the risks for which a solution will be proposed.

Risks that are obvious from the systems architecture prior to any formal walkthrough.
Risks that arise from the system-level CFIA table.
Risks that arise from the detailed 'node' analysis.
Risks that arise from previous problem records, and RCAs.

- **Resilience risks:** Risks which may cause a service to become unavailable in the first place. They are fundamental weaknesses.
  - Single point of failure in the IT infrastructure
  - No hot failover capability
  - Log files filling up
  - Bugs in code
  - Operator error
  - Old hardware
- **Recovery risks:** Risks which prevent the service from being recovered from an outage in a timely manner.
  - Responsibilities not defined clearly
  - Lack of, or incomplete recovery procedures
  - Lack of skills
  - Over-complex manual tasks with no automation
  - Recovery process is not tested
- **Security Risks:** Risks which can render a service to become useless, for example, through:
  - Security patch management
  - Privileged users who misuse their privileges
  - Denial of service attacks

RCA = Root Cause Analysis

(click to learn more)

For Resilience Risks, provide a risk rating for impact, occurrence, and detectability, using the rating scale in the Resilience Scale Table

Resilience Scale Table				
Rating		Business Impact (A node failure would ....)	Occurrence Time Period	Detect-ability (how easy to detect failure)
H	10	Injure a person.	More than once per day.	Only via a Customer complaint or notice.
	9	Create an illegal situation (e.g regulatory compliance)	Once every 3-4 days	Manual checking of individual nodes infrastructure by layered teams.
	8	Service unusable – performance or outage.	Once per week	Automatic monitoring & alerting on node infrastructure by layered teams.
M	7	Major customer escalation – performance related.	Once per month	Automatic monitoring & alerting on node infrastructure to a central team.
	6	Complaint - major performance degradation.	Once every 3 months	As above plus basic infrastructure correlation.
	5	Complaint –ongoing performance degradation.	Once every 6 months	As above plus basic synthetic transactions to defined points in infrastructure (e.g. data centre).
	4	Visible - minor ongoing loss of performance.	Once per year	As above plus real user transactions added (end to end).
L	3	Visible but can be overcome readily overcome	Once every 1-3 years	As above with extended application logging to assist root cause.
	2	Not visible - minor impact on performance.	Once every 3-6 years	As above with integration to an actively managed CMDB (accuracy).
	1	Not visible– no impact on performance.	Once every 6-100 years	Full business systems impact correlation of above (all nodes, all technical layers).

Assign a value on a 1-10 scale to each of:
- Business Impact: How severe is the failure.
- Occurrence: How likely is the outage to happen.
- Detect-ability: Can the cause be easily detected.

For each CI (Node, Component, Link, etc.), provide a risk rating for Teaming, Procedure, and Automation using the Recovery Scale Table

Recovery Scale Table				
Rating		Team (The team recovering the service)	Procedure (The procedures used are:)	Automation (The level of automation to assist recovery:)
H	10	Have little experience and no overall leader.	No procedures exist.	There is no automation in any aspect. Long manual recovery.
	9	Inexperienced teams can recover a few critical nodes, limited e-2-e team structure, no RM	Very few procedures exist and are not maintained or tested.	n/a
	8	Inexperienced teams can recover most critical nodes, limited e-2-e team structure, no RM	Some level of procedures exist but they are not maintained or tested	Infrastructure services recovered automatically, long and complex manual recovery of data/apps.
M	7	Inexperienced teams can recover all critical nodes in isolation, limited e-2-e team structure, no RM.	Most nodes have procedures but they are not integrated nor maintained/tested	n/a
	6	Experienced teams can recover all critical nodes, some end to end team structure, no overall RM	All nodes have procedures but they are not integrated or maintained/tested.	Infrastructure services recovered automatically, some well rehearsed manual recovery of applications or data. Intermediate outage.
	5	As below…some e-2-e structure	As above……Integrated, not maintained/tested.	n/a
	4	Experienced structured team with well defined e-2-e structure inexperienced RM	Integrated system level and node level procedures, Limited maintenance, no testing.	Infra recovered automatically with some application & data recovery automated. Minor outage expected.
L	3	As below…..RM has little experience	As below & …Limited Maintenance	Fully automated recovery with minor service outage.
	2	As below & ….RM has medium experience…	As below & …. Limited test	n/a
	1	Experienced and structured team with e-2-e service recovery managed by an experienced Recovery Manager (RM).	Integrated System level and node-level procedures exist and are regularly maintained and fully tested e-2-e	Fully automated recovery and will not be noticed by the Customer (e.g. hot failover).

Assign a value on a 1-10 scale to each of:

- Team (the team recovering the service)
- Procedure (the recovery procedures used)
- Automation (the level of automation to assist recovery)

For each CI (Node, Component, Link, etc.), provide a risk rating for Patch Management and, if applicable, for Privileged ID Management, using the rating scale in the Security Scale Table

Security (Availability related) Scale Table			
Rating		Patch Management	Privileged ID Management
H	10	Patch management is not undertaken.	user-ids shared amongst support staff with no continued business need or password management process.
	9	Patches applied ad-hoc to a few systems which are not critical to the business function.	user-ids not shared by support staff with no continued business need or password management process.
	8	Patches applied ad-hoc to a few, but not all critical systems.	user-ids not shared by support staff. Limited password management process/tool.
M	7	Patches applied manually across some systems but with no schedule or regularity.	user-ids not shared by support staff. Effective password management process/tool in place. No CBN Process.
	6	Patches applied manually across all systems but with no schedule or regularity..	user-ids not shared by support staff. Effective password management process/tool in place. Limited CBN Manual Process.
	5	Patches applied manually to all systems on a regular basis.	user-ids not shared by support staff. Effective password management process/tool in place. Limited CBN coverage. Partially automated.
	4	Patches applied manually to all systems, on a regular basis with strong procedural & verification testing.	Privileged user-ids managed. Effective CBN and full password control but not related to change management (e.g. Break-glass & Vaulting). Partially Automated.
L	3	Patches applied regularly to all systems in a secure and automated manner.	Privileged user-ids fully managed. Effective CBN and full password control but not related to change management (e.g. Break-glass & Vaulting). Automated.
	2	Patches applied regularly to all systems in a secure and automated manner. Patches tested and verified.	Privileged user-ids fully managed. Full password control related to change management (e.g. Break-glass & Vaulting). CBN less effective.
	1	The latest patches regularly applied ito all systems in a secure and automated way that conforms to a clear patch management policy. Patches veriried.	Privileged user-ids fully managed. Effective CBN and full password control related to change management (e.g. Break-glass & Vaulting). Fully automated.

Assign a value on a 1-10 scale to each of:
- Patch Management
- Privileged ID Management

Areas that might be considered for improvement

- Identification of Single Points of Failure (SPOF), where loss of a single component would impact on the non-functional characteristics of an IT service.
- Application Design, for example singleton processes to read data queues that represent single points of failure, and Application Integration where many systems are tightly coupled together as part of an IT service
- Missing or inadequately documented architecture for the IT service and operational procedures.
- Performance, capacity or scalability concerns for any component, in response to increases in demand.
- Monitoring may be missing or otherwise deficient, resulting a component or service outage not being detected (eg. No monitoring of overall IT service availability).
- Deficiencies in backup and restoration procedures that may impede recovery operations (eg. Data being stored on lots of separate tapes or backup data not being held securely).
- Processes and procedures to recover from a failure (or failover) are missing or deficient. Manual intervention means much more significant delays in recovery.
- Poor change control procedures, where changes are made to an IT service without adequate risk assessment being conducted prior to deployment. This might consider the nature and comprehensiveness of testing (eg. Does it include ,), adequacy of back out procedures and whether changes made in other systems might impact on the IT service being considered as part of this.
- Management of the configuration of hardware, system software, middleware and applications is consistent across development environments (eg. Development, Test, UAT, Pre-production and Production) and across clustered servers in the same environment (eg. Multiple clustered Web servers)
- Deficiencies in Testing procedures or the test environment itself (i.e. the system can't be tested under "production like" conditions)
- Adequacy of backup and restore procedures
- Technical Inadequacies in the solution such as software approach or beyond end of life or use of unsupported software combinations,
- "Key Person" dependencies – where a single individual is responsible for technical support or operations for one or more components essential to the successful operation of the IT service.