SPA2007 - WS1
Strategies and Patterns for Systems Continuity

Eoin Woods and Nick Rozanski

Example System 1 – Financial Accounting

System Overview

A retailer is implementing a package (which has already been pre-selected by the sponsor, the CFO) for financial accounting and management.

The chosen package, ZAP Finance, will provide the following key functions:

- purchase orders and invoices
- automated transaction matching
- payments
- statements
- reference data management (product, supplier etc)
- some logistical support (shipments, warehousing stock)
- financial control and management information.

ZAP Finance will be used to manage both merchandise (product sold by the retailer) and non-merchandise (services and materials used by the retailer). It will be used internally and some features will be made available to trading partners.

An overview of the system is shown in Figure 1.

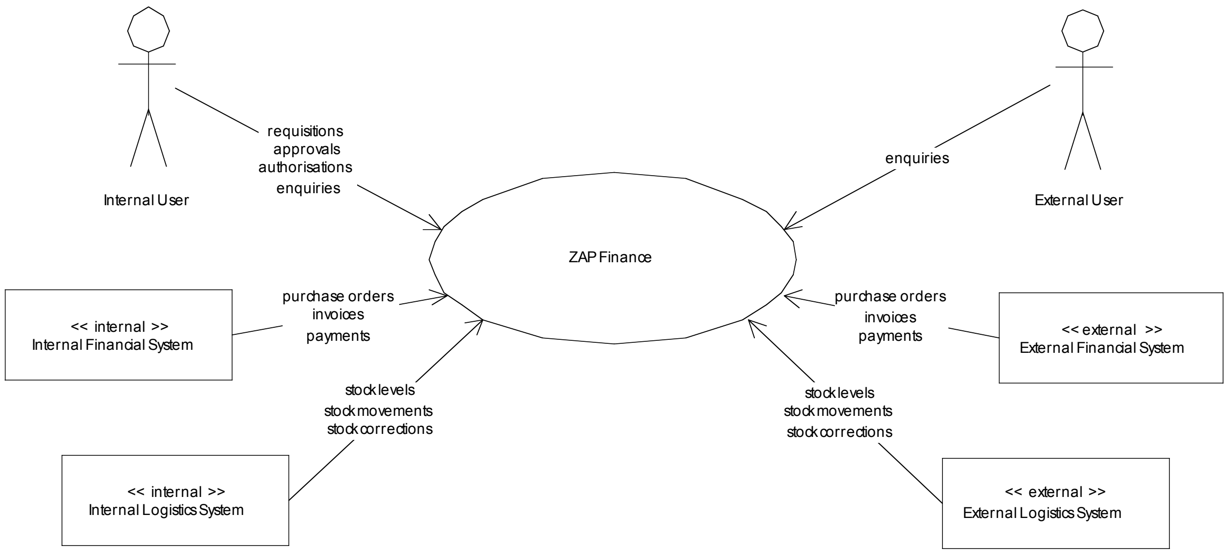

Figure 1 - Financial Accounting System Context

The main entities that interact with ZAP Finance are as follows:

- *Internal Users*. These include Finance senior management (including the CFO), operational staff from the Finance Department, and a small number of other users throughout the company. The internal user population numbers about 50.
- *External Users*. Query access will be provided initially to a small number of suppliers, trading partners and other third parties. The number of external users is small, but may grow substantially in the future.
- *Internal Financial Systems*. These provide supporting functionality for processing and analysing purchase orders, invoices and payments. There is a large number of such systems (over 30).
- *Internal Logistics Systems*. These manage logistical information such as stock levels, shipments and other stock movements.
- *Supplier Financial Systems*. These receive purchase orders from ZAP, submit invoices to ZAP and participate in payment transactions.
- *Supplier Logistics Systems*. These send and receive logistical information such as shipments and returns.
- *Internal Treasury Management System*. This system receives key financial data from ZAP for further processing, and makes daily submission to HM Treasury.

The key "non-functional" requirements of the system are as follows.

- It must be able to scale and "flex" in line with the fortunes of the business.
- It must authenticate and authorise all internal and external users to an appropriate level. Strong authorisation is required for key users to control access to sensitive financial data.
- It must strongly authenticate all external interfaces, particularly those involving financial transactions.
- It must allow user operational access during standard working hours, with extended access (evenings, weekends) being required at key times in the financial calendar such as quarter- or year-end.
- The Recovery Time Objective (RTO) seems to be <u>24 hours</u> in the event of a disaster. (In practice, much of the system could be unavailable for longer than this, but the Treasury reporting functionality <u>must</u> be up and running within this time.)
- Data is extremely valuable and the goal (this may not be entirely achievable) seems to be no or minimal data loss in the event of a disaster – that is, an RPO <u>near zero</u>. In particular, the regulatory reporting data must be fully recovered.
- The "usual" NFRs for a system of this nature apply, such as "no single points of failure."

Functional View

Functionally, ZAP is relatively straightforward, as shown in Figure 2.

Operational and management users access the system from their desktops (using a thin or rich client interface) or from an external location over a secure link. A common application tier performs the requested processing and manages the financial data held in a central database.

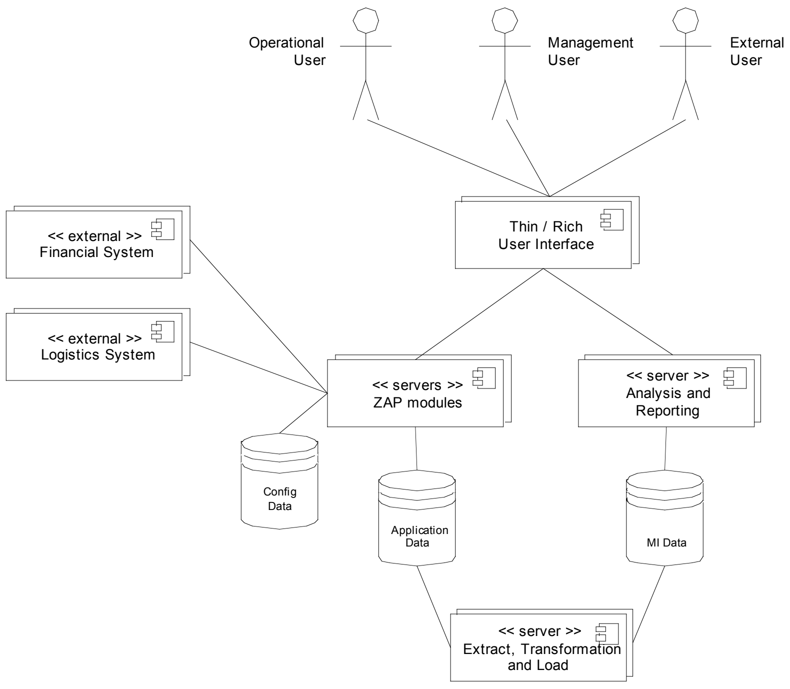
Operational User
Management User
External User
Thin / Rich User Interface
<< external >> Financial System
<< external >> Logistics System
<< servers >> ZAP modules
<< server >> Analysis and Reporting
Config Data
Application Data
MI Data
<< server >> Extract, Transformation and Load

Figure 2 - Financial Accounting Functional Model

The application tier comprises a number of different modules as follows:

- **Accounts Receivable** records merchandise and non-merchandise account postings generated from customer sales and other payments received and updates the General Ledger.
- **Accounts Payable** records merchandise and non-merchandise account postings generated from supplier purchases and other payments made. It also updates the General Ledger.
- **Asset Accounting** manages the company's fixed assets. Assets are categorised by class, with each asset class having its own depreciation characteristics.
- **Bank Accounting** manages bank transactions, including cash management.
- **General Ledger** records all account postings and provides real-time visibility of the financial accounts.
- **Internal Orders** tracks the costs and business transactions associated with internal jobs, services and tasks such as store replenishments.
- **Profitability Analysis** helps management to understand profit or contribution by business area.
- **Treasury Reporting** generates, manages and submits a range of reports required for regulatory purposes. This data is further processed by another system before submission to HM Treasury.

An analysis and reporting component extracts financial data from the database and loads it into a data warehouse from which a large number of management reports are generated (viewed and / or printed in hard copy).

Internal and external systems are integrated with ZAP by means of batch and real-time integration modules. These are responsible for receiving or sending data, and for transforming it between ZAP format and the data formats used by other systems.

The behaviour of the application tier is strongly customiseable, with the customisation information being managed in a separate configuration data store.

Information View

The information held in ZAP Finance is of key importance to the business. It comprises three primary datastores:

- The **Application Data Store** manages all of the application data. It includes the chart of accounts, ledger postings and other transactional data, derived financial data and reference data. This data is manually entered or automatically loaded into ZAP from external systems, usually in batch. The Application Data model is extremely complex, comprising many hundreds of entities.
- The **Management Information Data Store** is a data warehouse which manages data extracted from the Application Data Store. This data is transformed and aggregated for comprehensive analysis and reporting.
- The **Configuration Data Store** manages the ZAP configuration data. This data (which is set up as part of ZAP implementation) defines the behaviour of the customiseable elements of ZAP. It is managed by developers under strict change control, with no user access.

ZAP exchanges a large amount of data with external systems for logistics and financial management. (External in this instance means the retailer's own systems outside ZAP, and systems managed by suppliers and other third parties.) In most cases ZAP is regarded as the "data owner" – that is, it is the authoritative source of financial data – and such data can only be changed through ZAP.

A high degree of security is necessary around some of the data, especially the Management Information Data, which is confidential and can be highly sensitive during certain periods in the financial calendar (end of quarter, end of year etc).

Data volumes are fairly predictable and grow steadily in line with the growth of the business. Once historical data is taken into account, the Application Data Store is expected to grow to 100TB and the Management Information Data Store up to ten times as large (volumes to be confirmed). In addition to volume increases at period end, there are other significant variations in the volume of data processed, which reflect the seasonal nature of the business.

Deployment View

A number of key decisions are yet to be made regarding the deployment of the system. These decisions will be highly influenced by the business continuity architecture for ZAP. Current thinking is as follows.

- The ZAP Finance system will be deployed on a high-end IBM AIX (Unix) platform. The implementation technology is primarily based around Java technologies and the database back-end is IBM UDB (DB2). These technologies are pretty much dictated by the ZAP product vendor (an analysis has been done, based on projected volumes, to choose a suitably resilient and scalable platform).
- All stateless servers will be load-balanced. In the event of a failure, the failed node will go off-line and the workload will be processed by the remaining operational node. Note that the drop in processing capacity could be a significant issue at period end.

- Stateful servers (ie the database servers) will be clustered using IBM's clustering technologies, which are provided as standard by the platform. The clustering model is "active-passive" – in the event of failure, a replica server will start on another AIX LPAR and client connections will be automatically transferred to this server. Some in-flight data may be lost during this process, which takes only a few minutes.
- All data (database, file etc) is held on a highly resilient and performant SAN (Storage Area Network). SAN connectivity is implemented over a high-bandwidth, resilient, virtualised fibre channel infrastructure within the data centre.
- ZAP is connected to end user desktops over the company's resilient WAN (Wide Area Network). Connectivity for remote internal users and external users is via a number of VPNs (virtual private networks) of varying quality and bandwidth.
- All of the system's data will be backed up to disk and then to an off-site location managed by a third party.
- Development, quality assurance and pre-production environments are also required. In the event of a disaster, a limited development / pre-production environment is required in order to build and deploy emergency fixes.

The company is investigating the use of a dedicated or shared disaster recovery site which would be activated in the event of a disaster or significant systems failure. No decisions on how this would work have yet been made, nor as to how workload might be allocated across the two sites.

Other internal systems with which ZAP Finance has to interface are implemented on a range of platforms, including the mainframe, several Unix flavours, iSeries (AS/400) and Windows. These systems have varying degrees of resilience and disaster recovery capability.