

coding {the} architecture



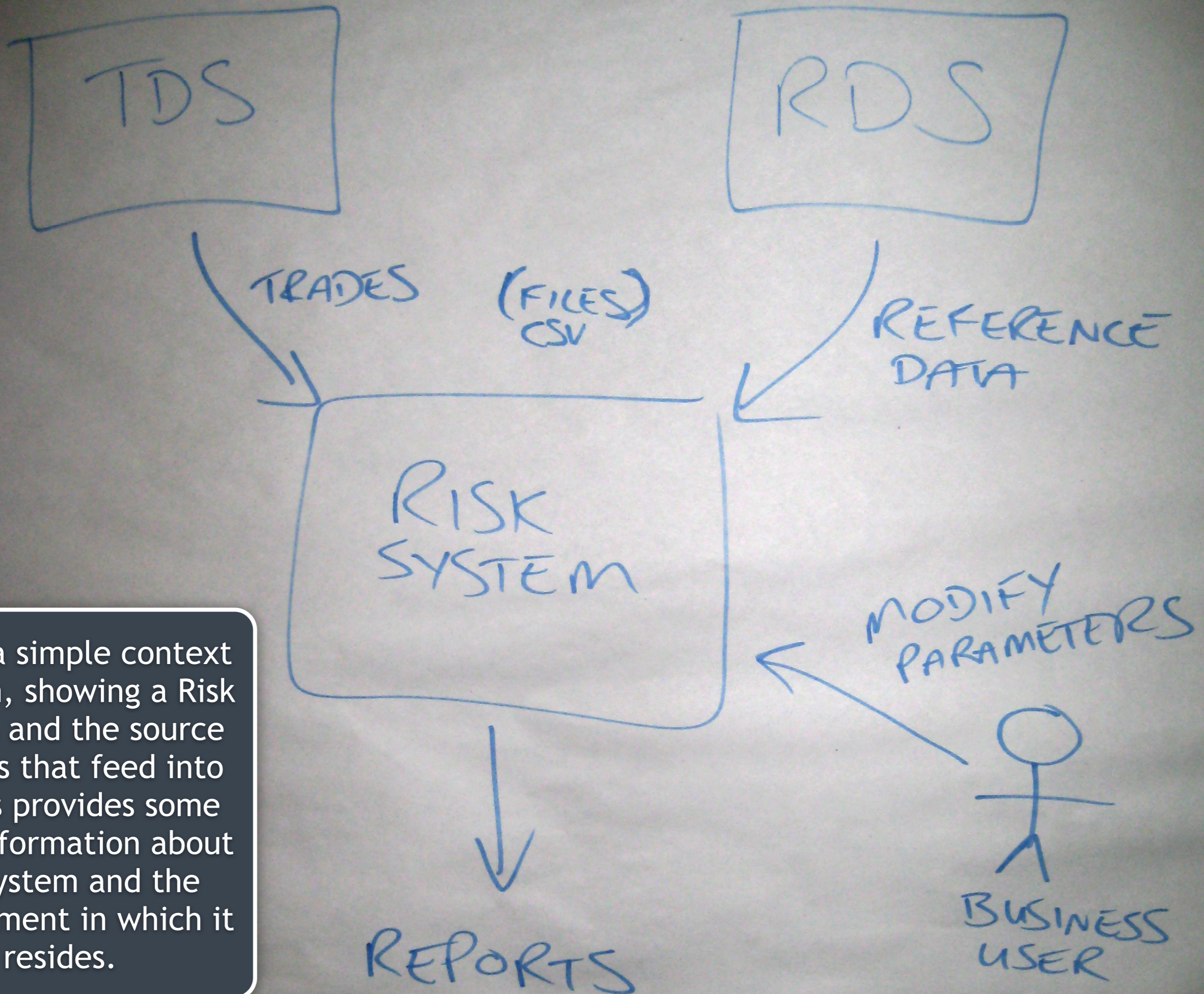
Example architecture diagrams

The diagrams here are some examples of the diagrams regularly used to document software architectures.

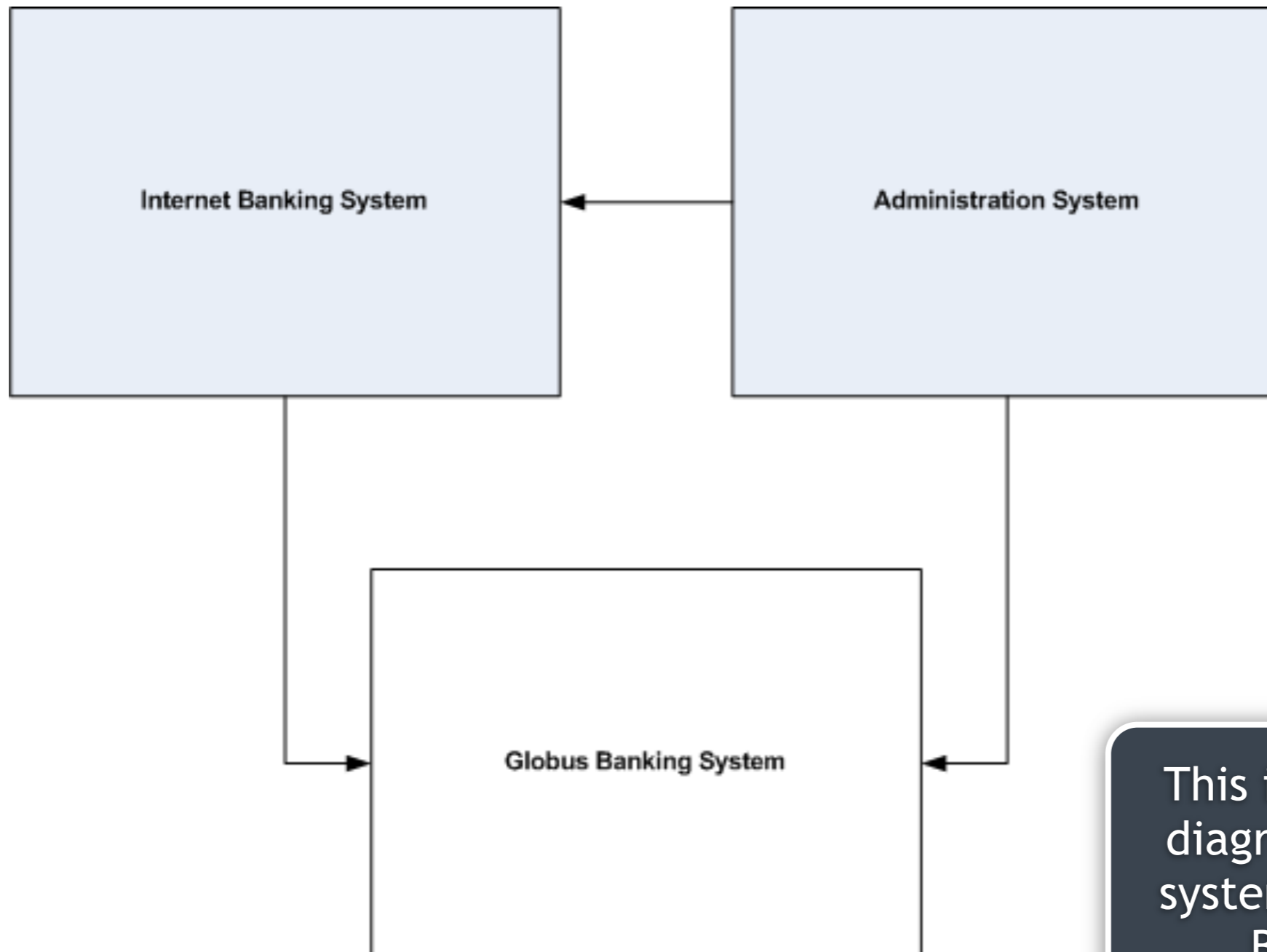
Some are taken from our “From Developer to Architect” training course (where groups are asked to define an architecture from a set of functional and non-functional requirements for a financial risk system) and some are from real projects. The diagrams illustrate how a system can be described from different views, along with the differing level of detail and notation.

Context

Context diagrams allow you to
set the scene



This is a simple context diagram, showing a Risk System and the source systems that feed into it. This provides some basic information about the system and the environment in which it resides.



This is a Microsoft Visio block diagram that summarises the system context of an Internet Banking System (plus administration system). It shows relationships with the existing Globus Banking System.

Functional View

Functional diagrams summarise what the
system actually **does** and helps you
show the **architecturally**
significant use cases



This is a high-level UML use case diagram that summarises what an Internet Banking system does at a functional level. It shows the key functional areas and how the functionality is utilised by two key types of users; customers and administrative staff.

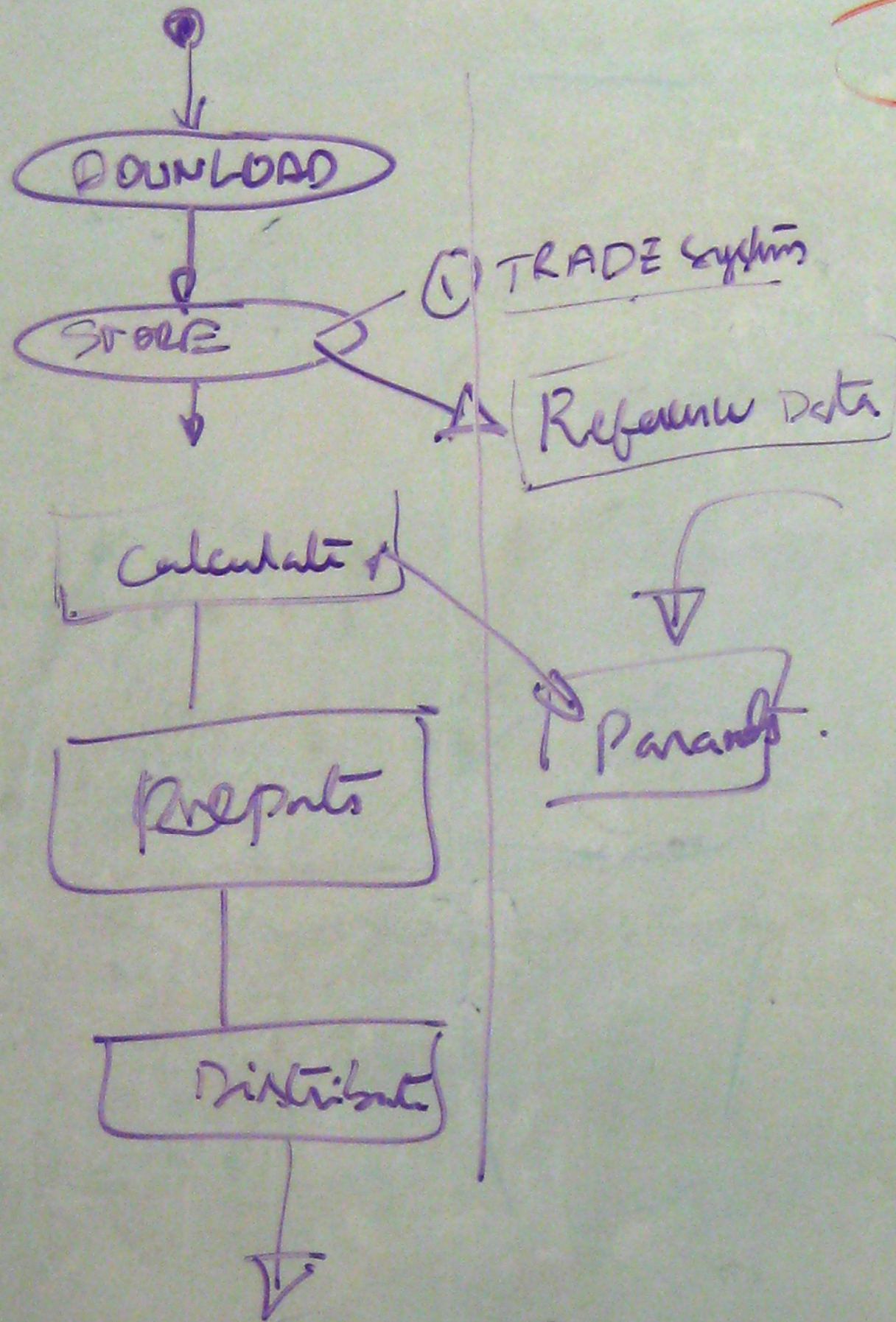
Process View

Process diagrams can be a useful
way to summarise the

overall process

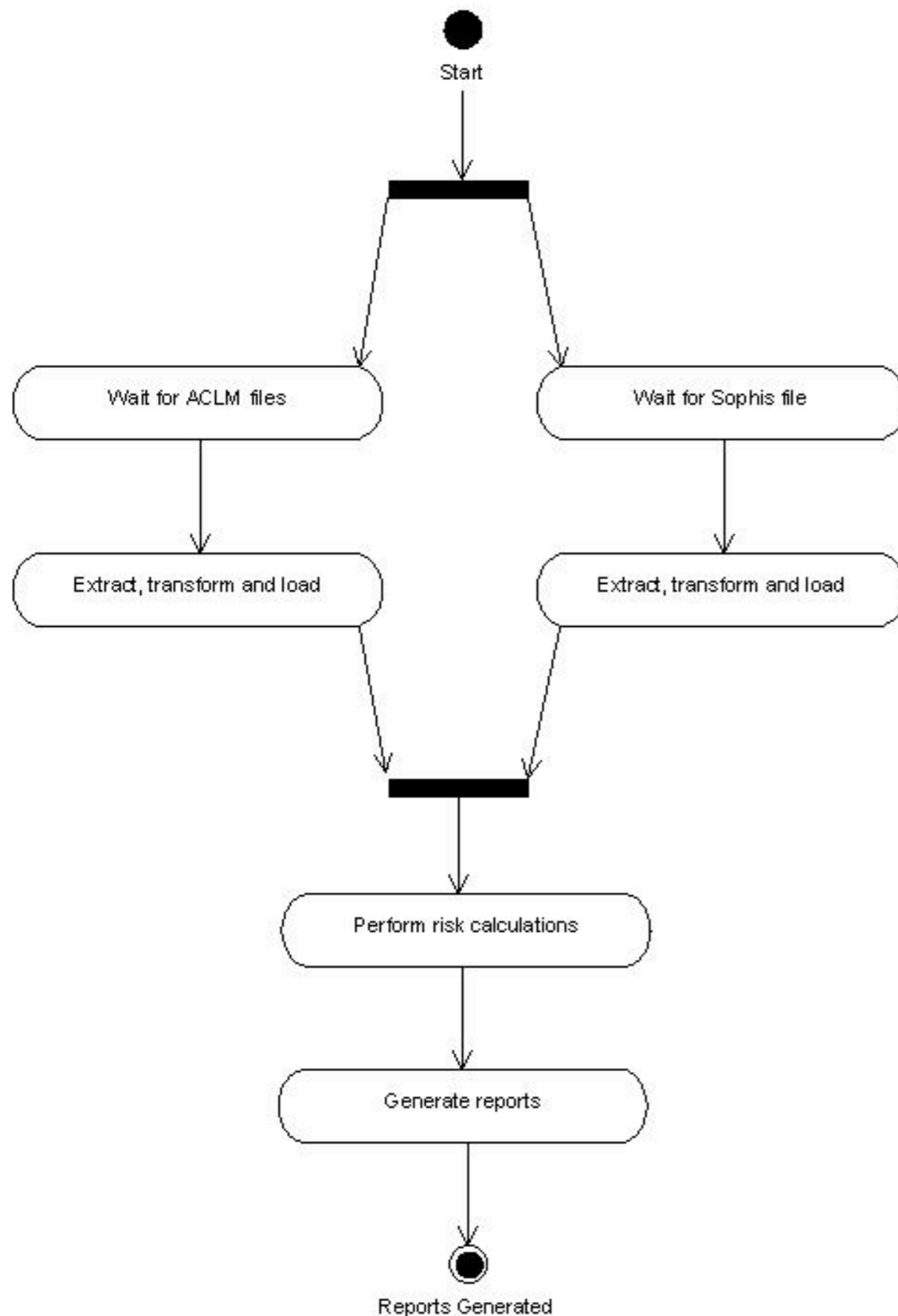
being implemented by the system

(steps, their order, flow of information, concurrency and parallelism)



This diagram shows the overall process executed by a system; from files being downloaded from the source systems through to reports being distributed.

It's often useful to have a high level process view because it helps understand the problem and is a good starting point for the high-level logical architecture.

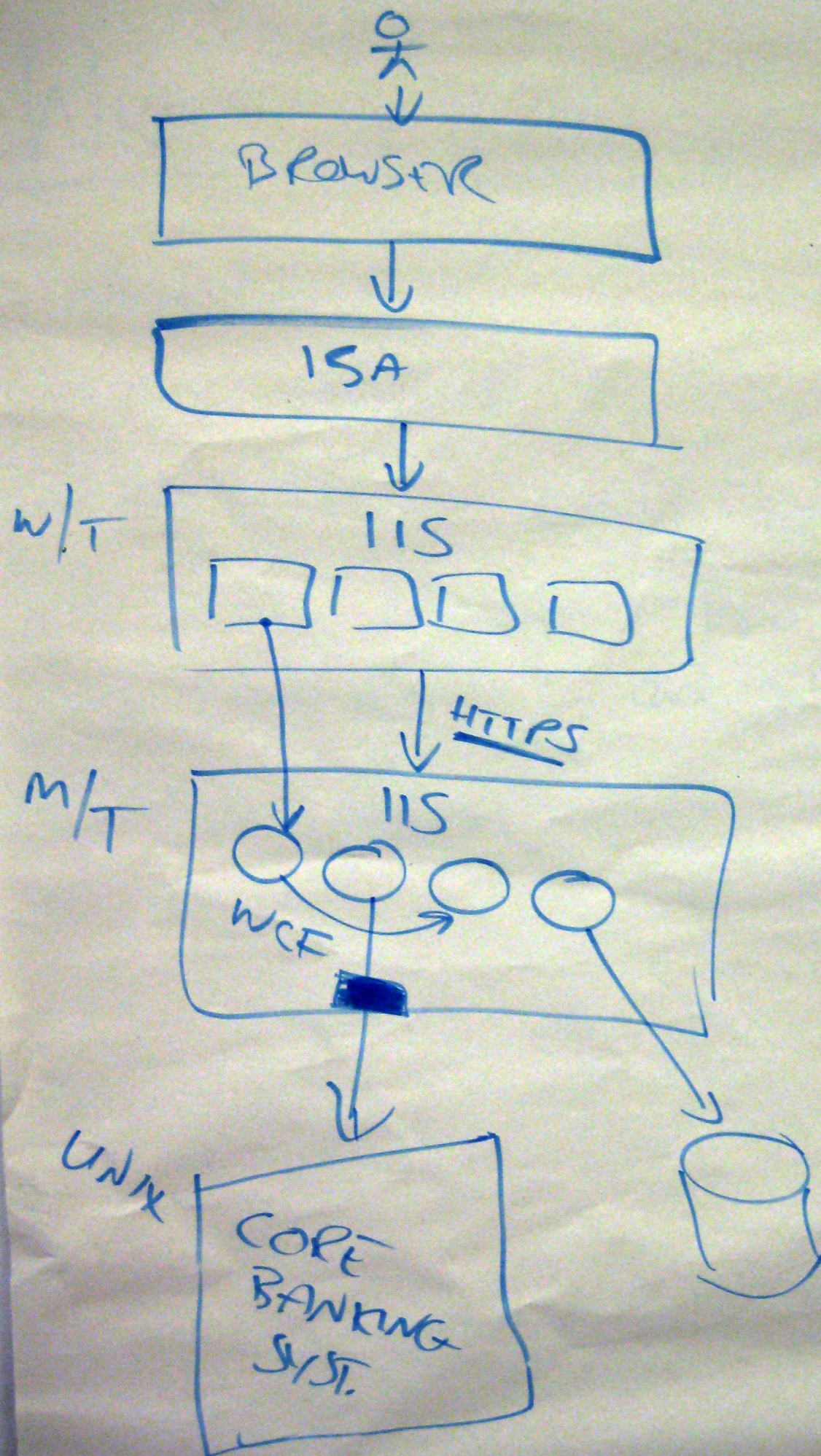


This is a UML activity diagram that summarises the process being implemented by a system. It shows the important steps along with the sequencing and where parallelism can occur.

Logical View

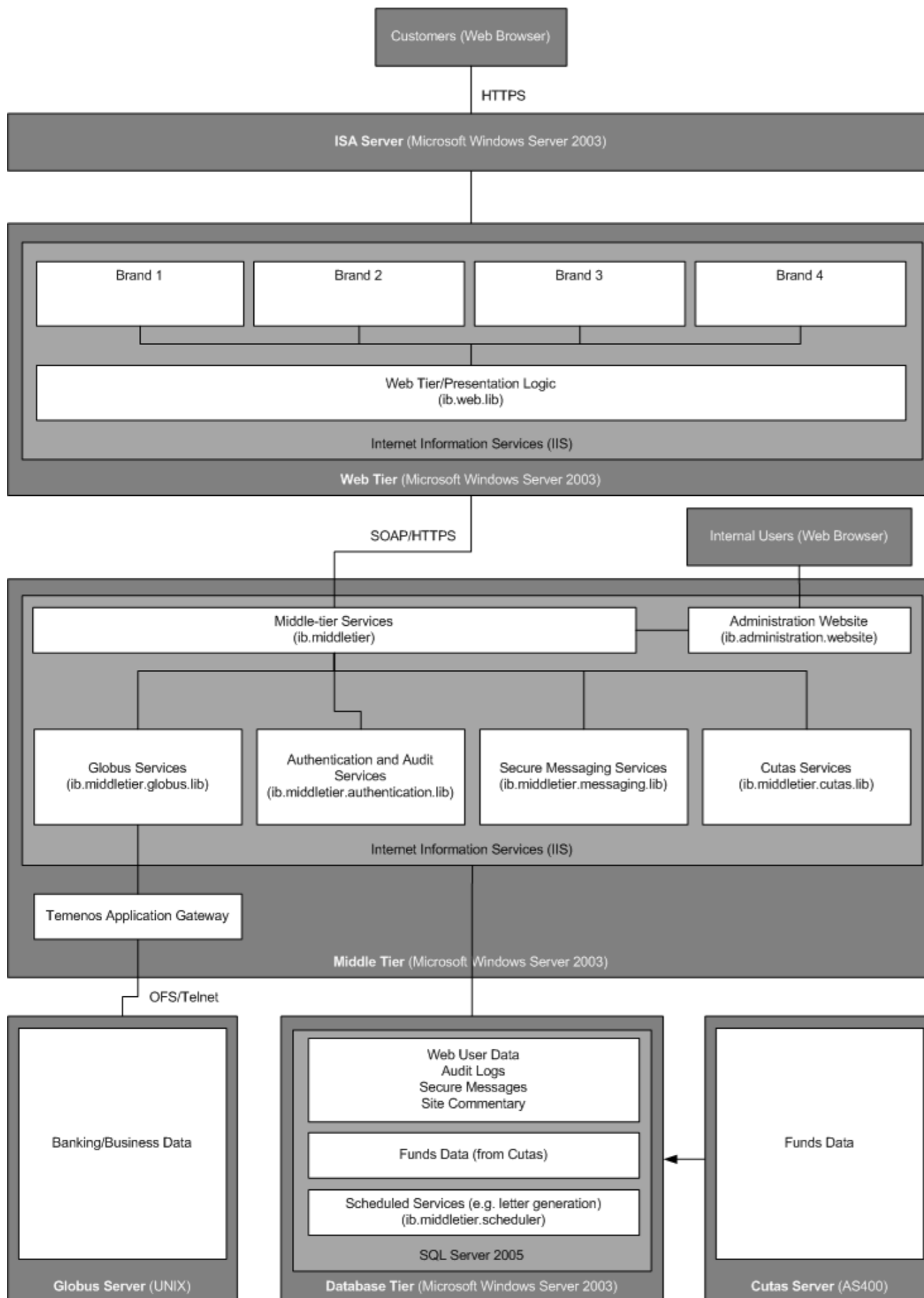
Logical diagrams show the
major components
and their **interaction**

(including technology choices)

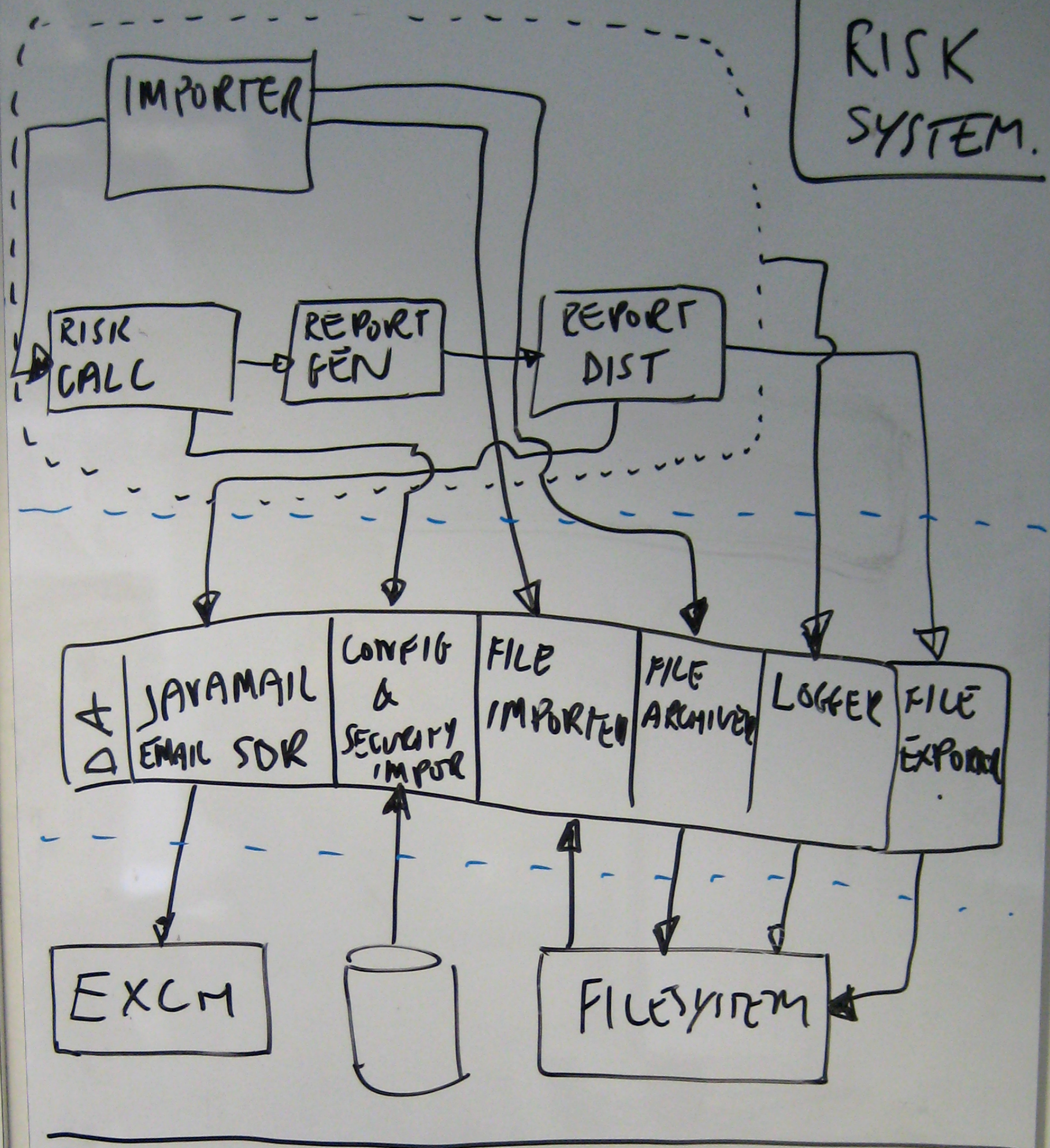


This is a drawing that summarises the logical architecture of an Internet Banking system. It shows the major architectural tiers and their interactions.

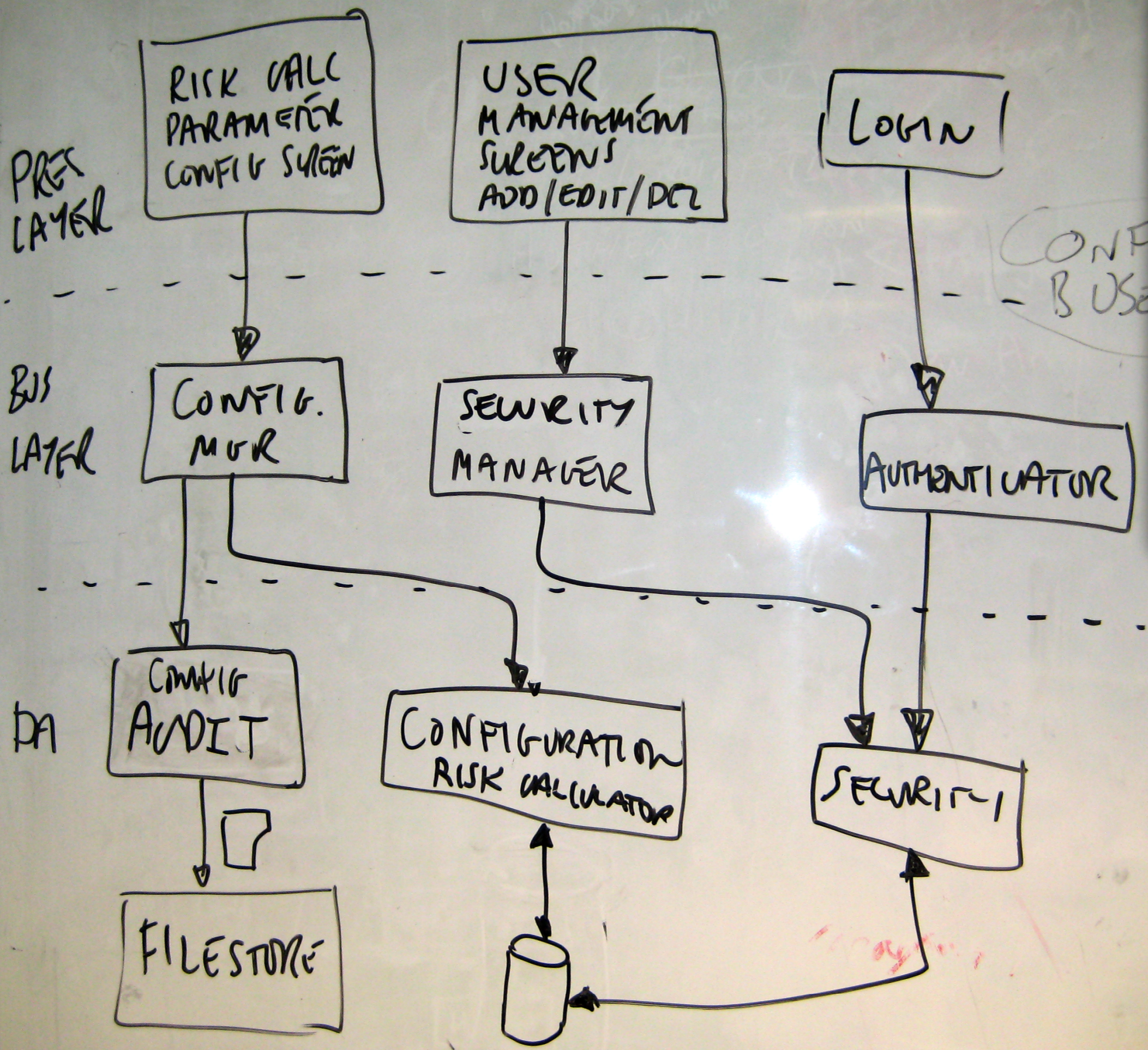
It's the sort of diagram that many people would use to start explaining how a system is designed and how it works, forming the basis for further diagrams with more detail.



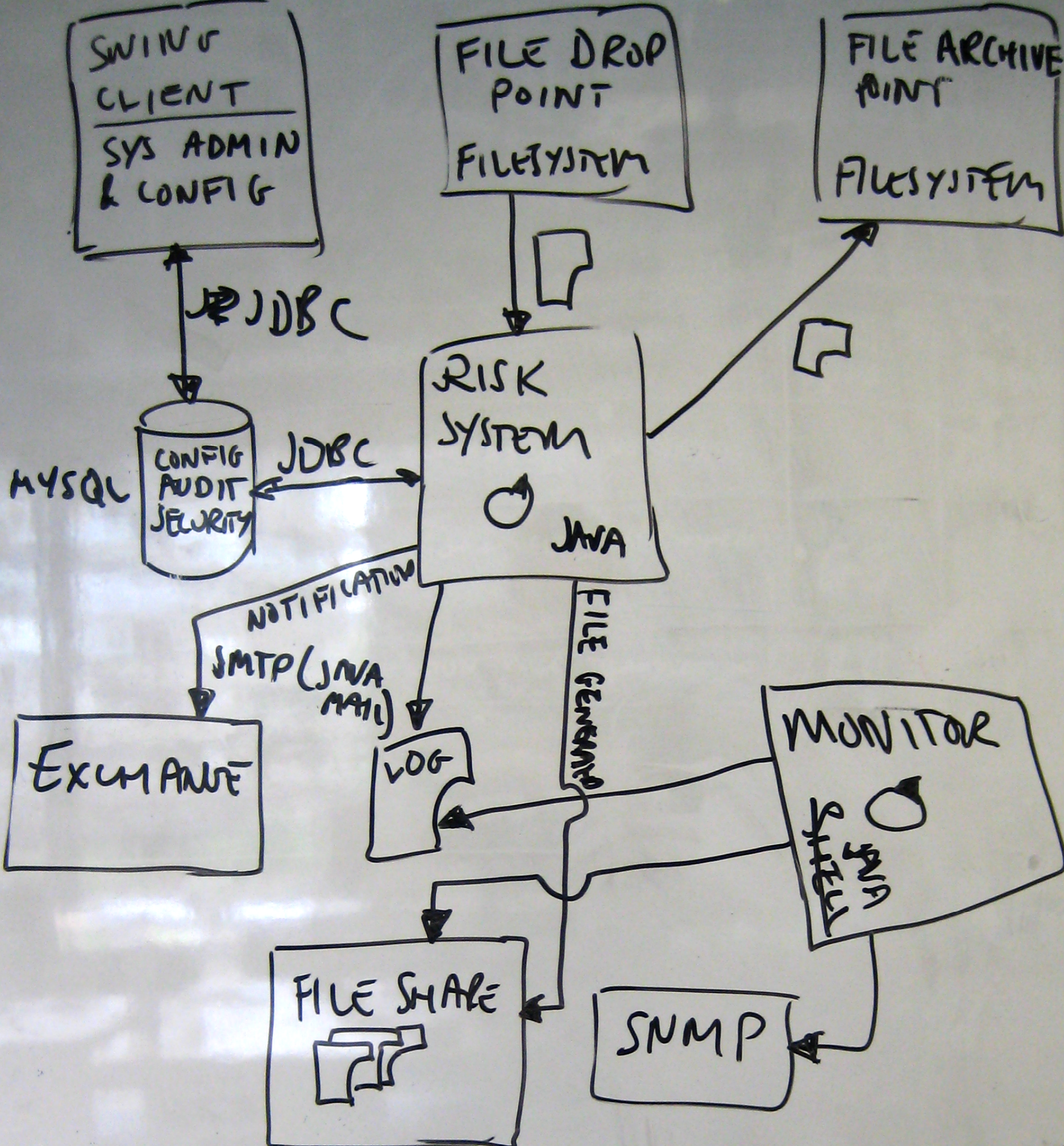
This is a Microsoft Visio block diagram that summarises the logical architecture of an Internet Banking system. It shows the major components and their interactions across all tiers of the architecture. It also includes information about technology, although would need to be supported with some commentary for each element on the diagram.



This diagram shows the overall architecture of a financial risk system, broken down into the major logical components. The dotted lines indicate the architectural layering adopted.



This diagram shows the logical layers of the architecture for a financial risk system and where the major components reside. It focusses on the user interface part of the solution only.

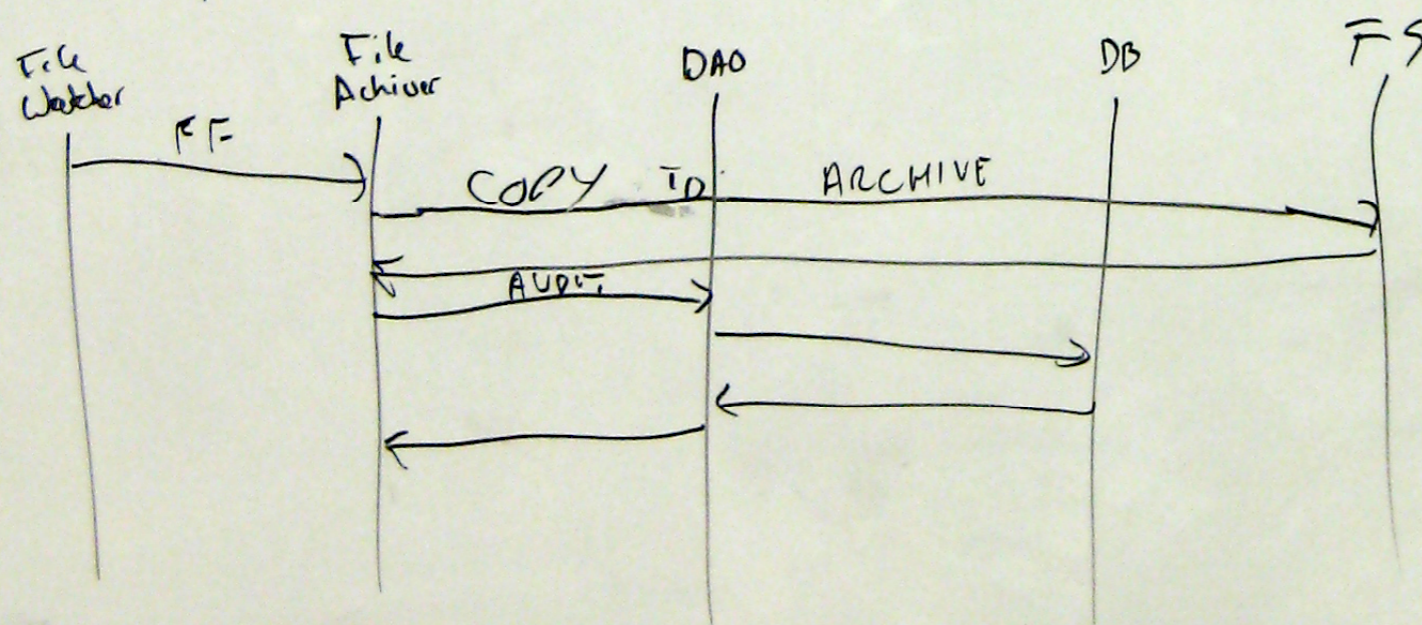
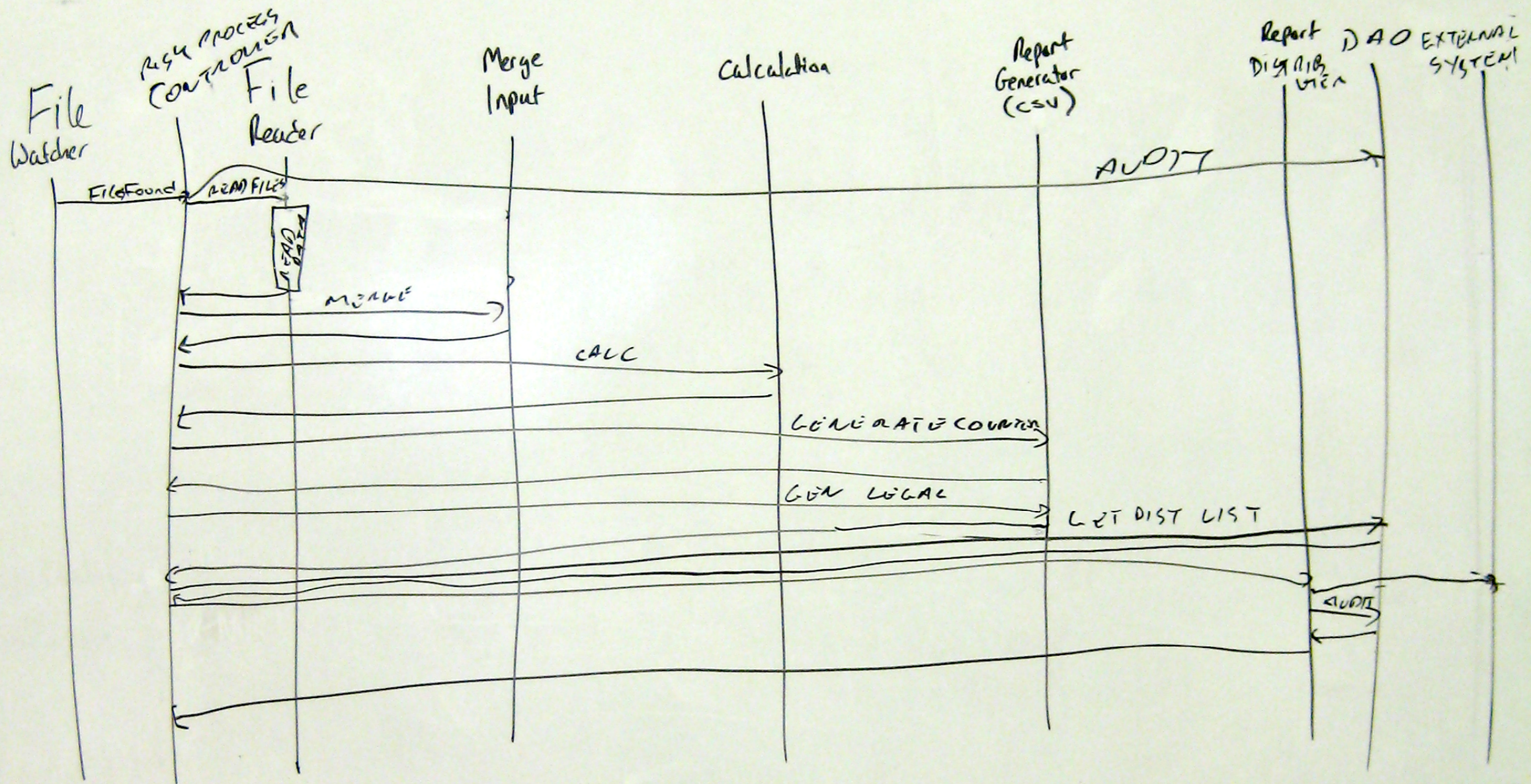


This diagram shows the overall architecture of a financial risk system, broken down into the major logical components at a high level of abstraction. It also shows some initial thoughts on technology choices.

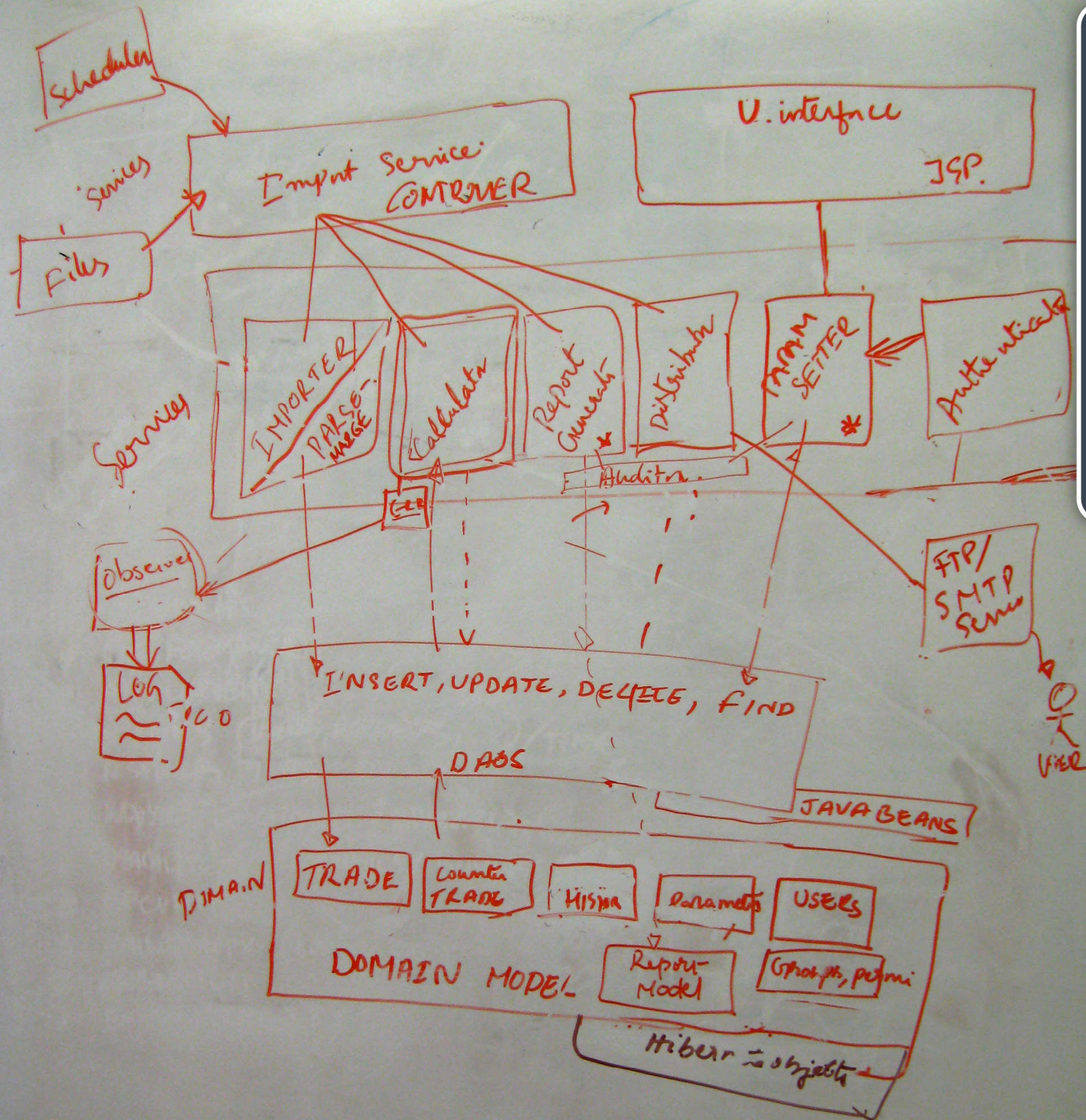
DA
E

EX

mo

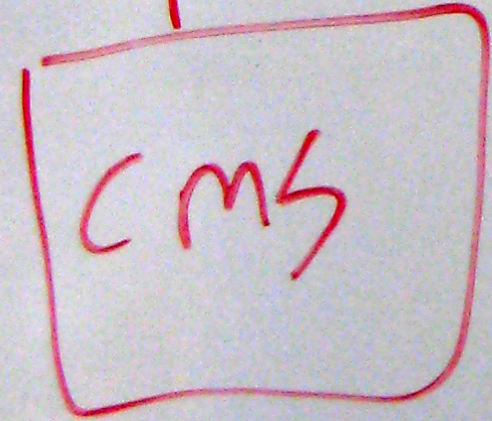
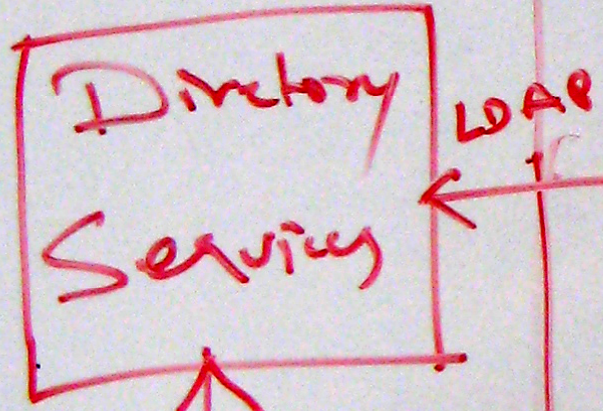
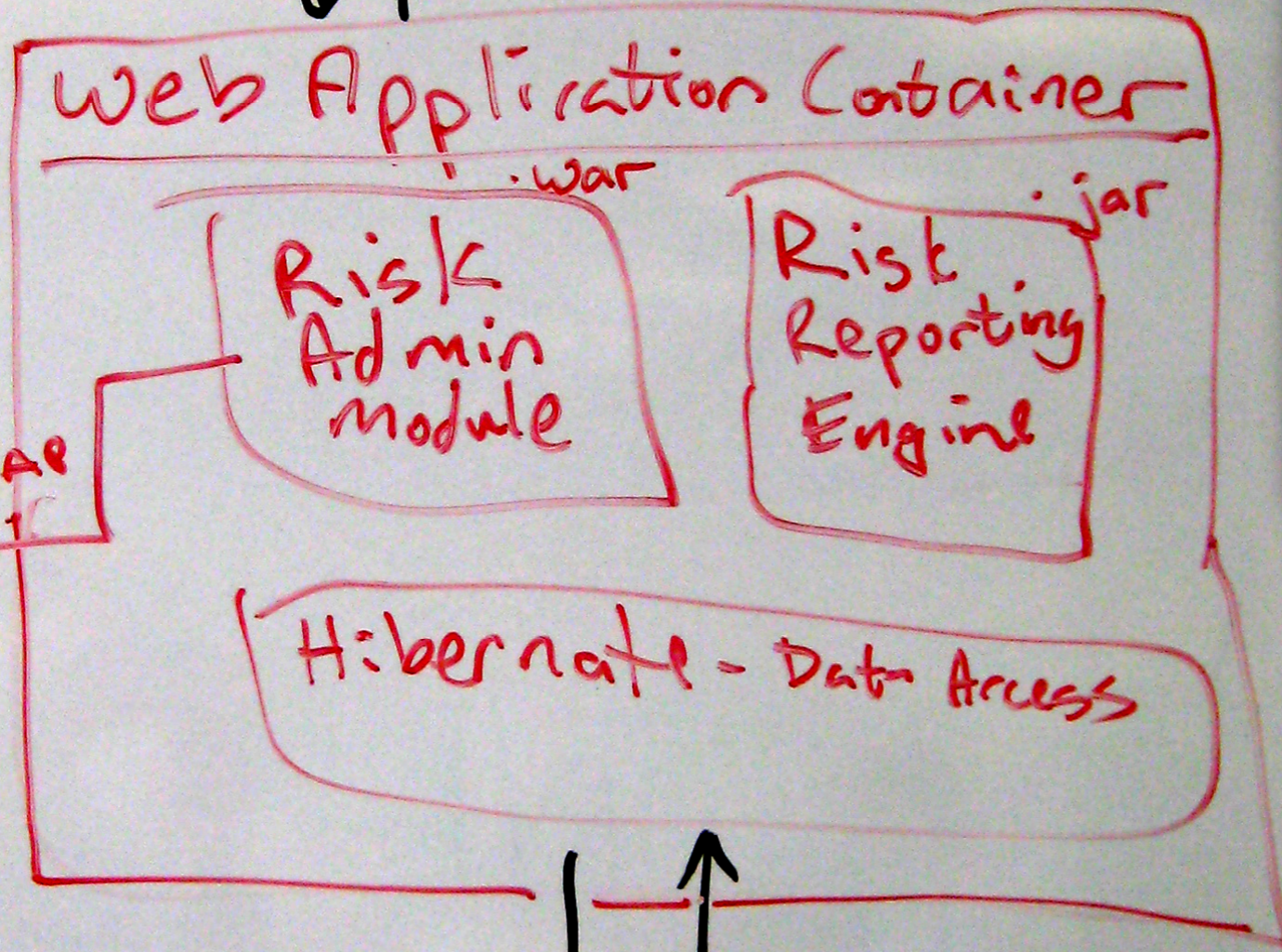
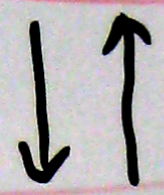


These sequence diagrams show the interactions between major components for key use cases of the financial risk system.



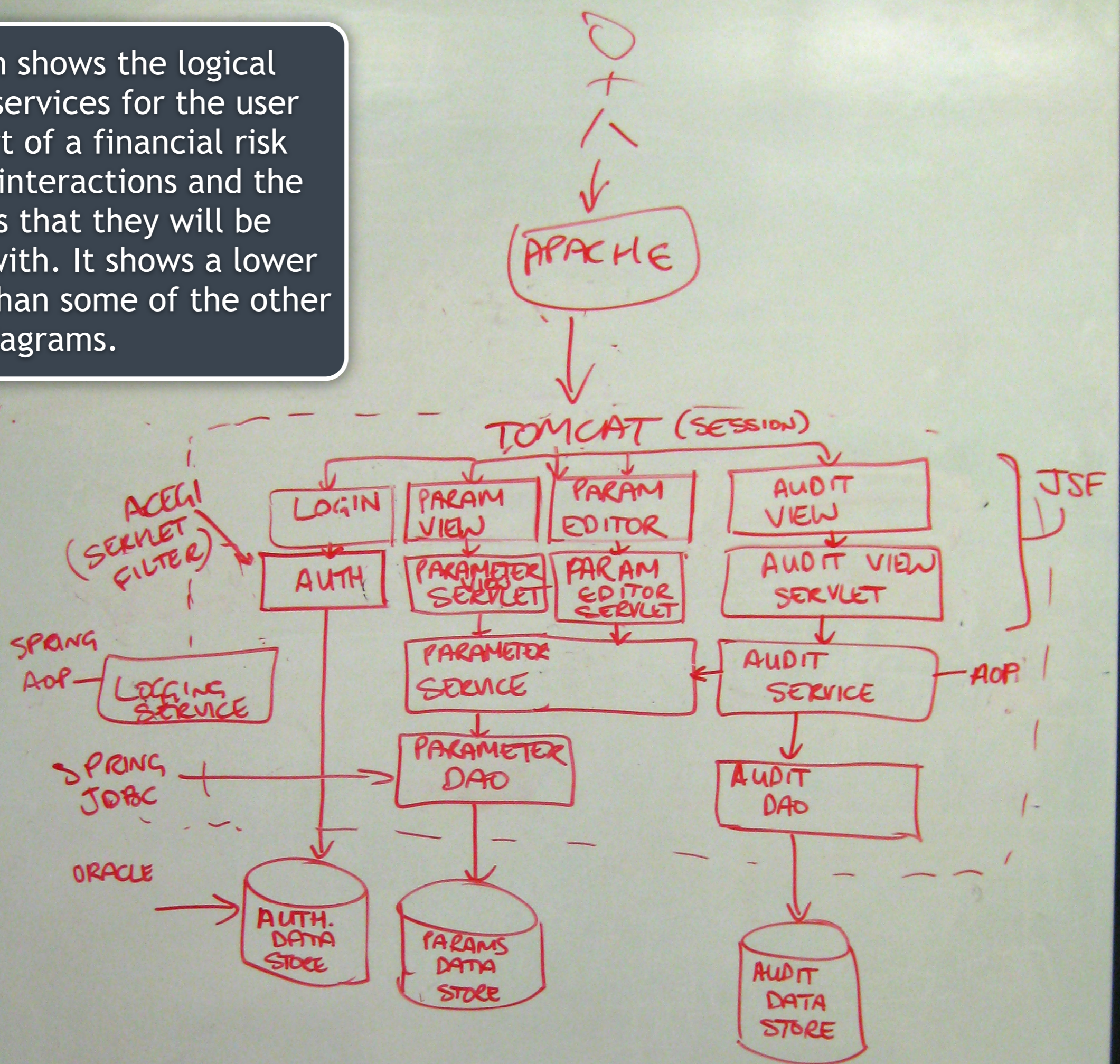
This diagram shows the overall architecture of a financial risk system, broken down into logical layers with two interfaces (scheduler and UI). The scheduler is the initial stimulus for the overall process.

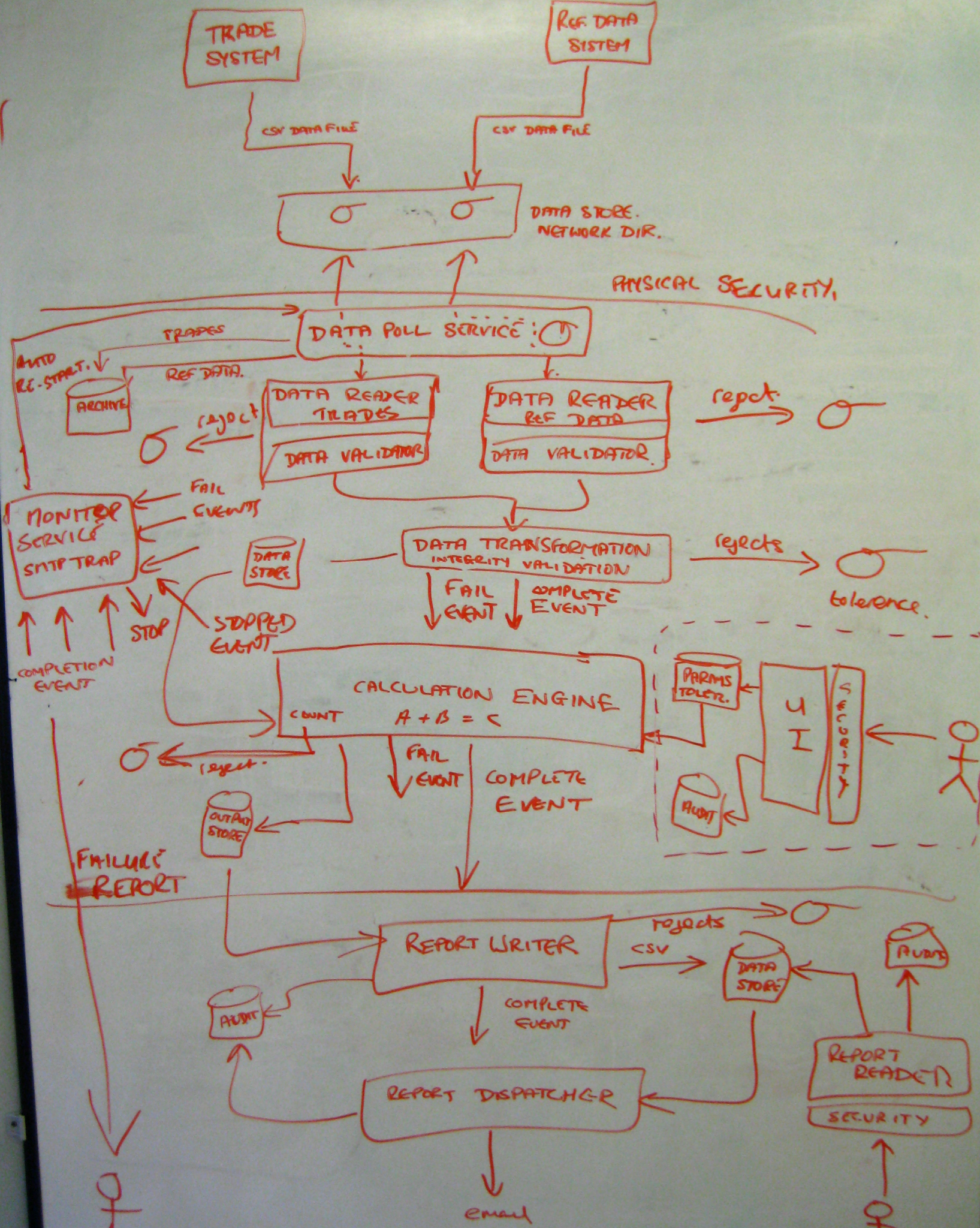
Web Browser



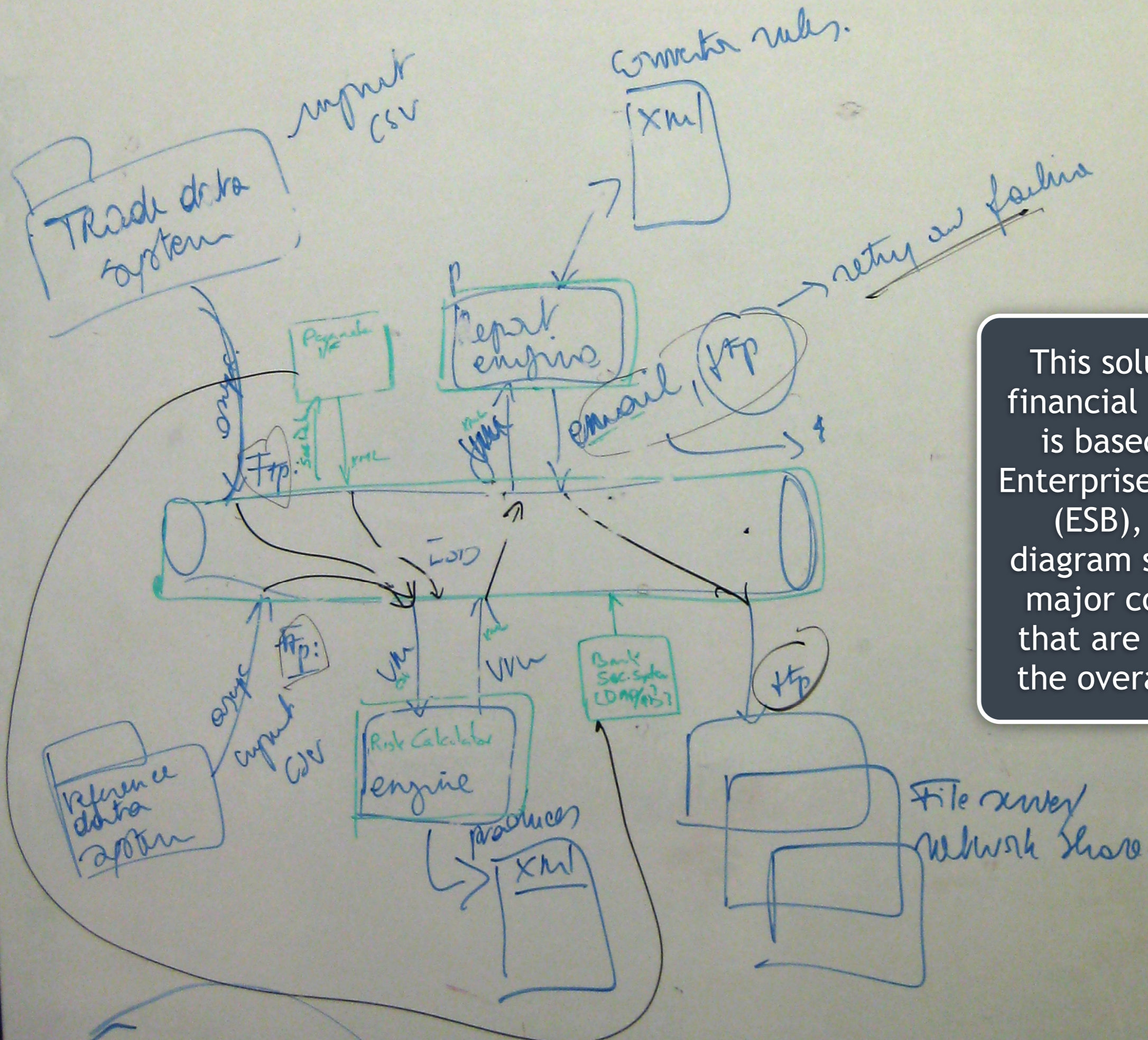
This diagram shows the overall architecture of a financial risk system, broken down into the major logical components at a very high level. It focusses on the user interface parts of the solution, but from a technology perspective.

This diagram shows the logical components/services for the user interface part of a financial risk system, their interactions and the technologies that they will be implemented with. It shows a lower level of detail than some of the other diagrams.

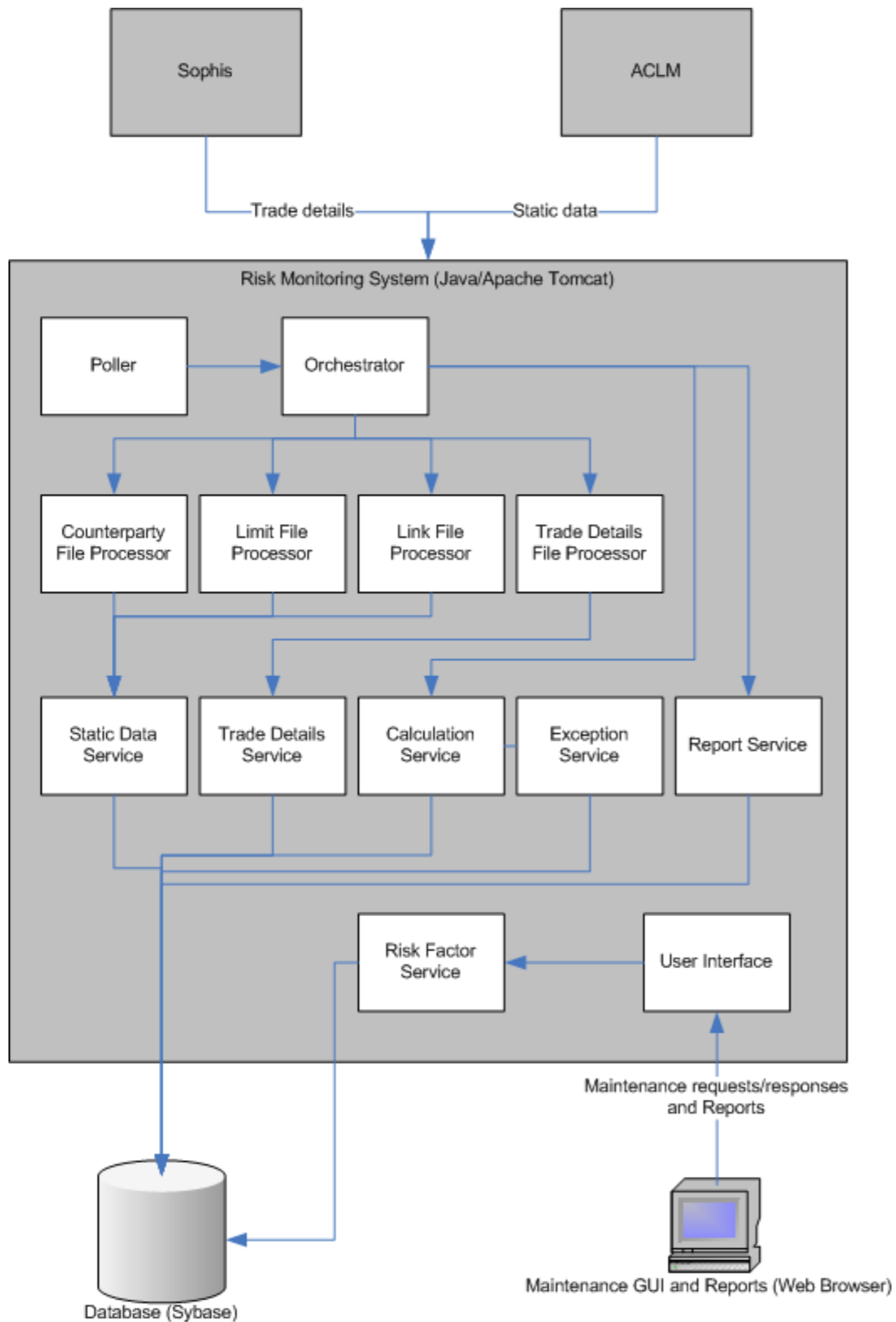




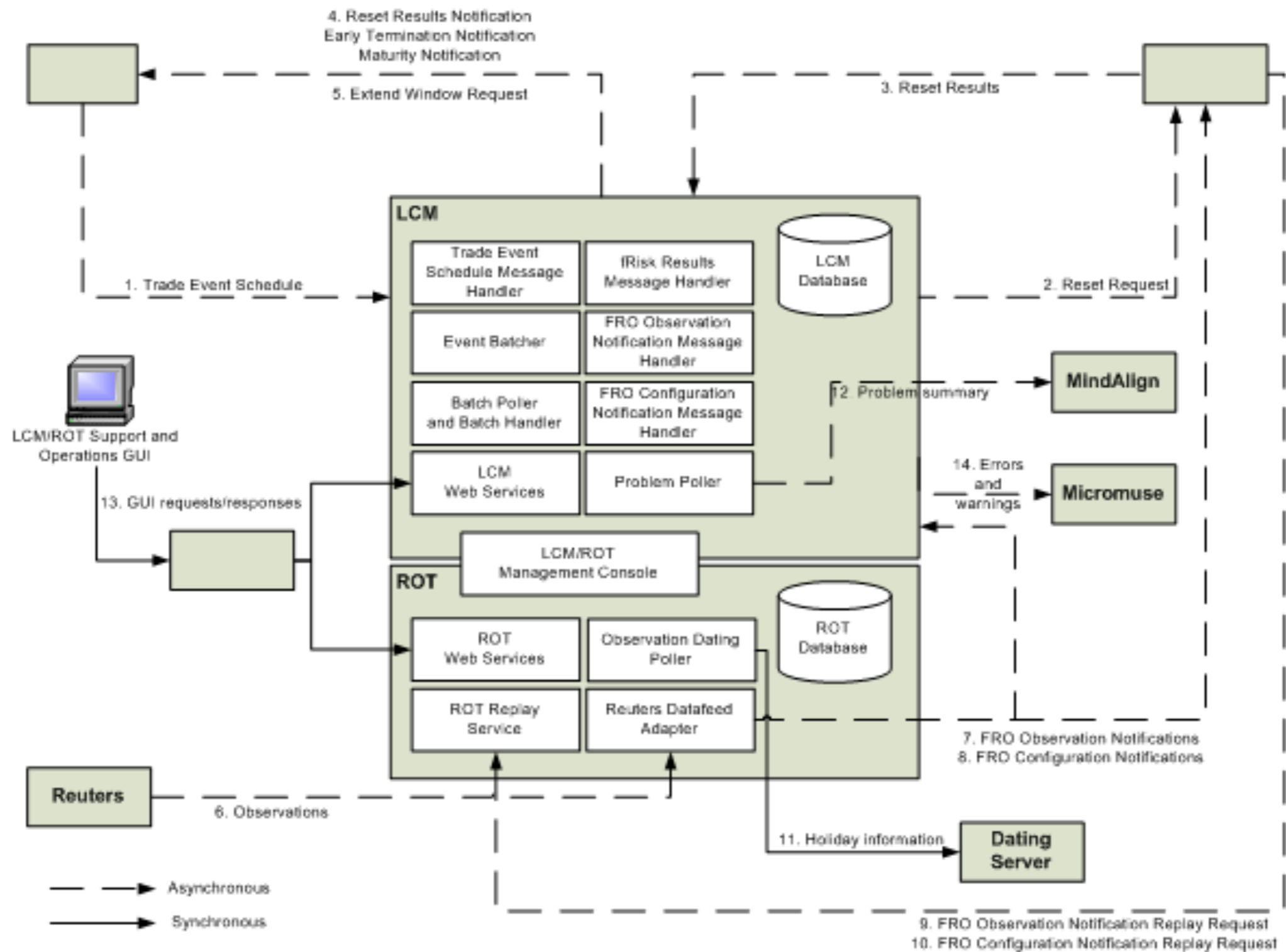
This architecture, for a financial risk system, is event based and the diagram shows each of the major logical components along with the events that they generate/consume. It also works on a polling rather than scheduled



This solution for a financial risk system is based upon an Enterprise Service Bus (ESB), with the diagram showing the major components that are involved in the overall process.



This is a Microsoft Visio block diagram that summarises the logical architecture of a financial risk system. It shows the major components and their interactions.



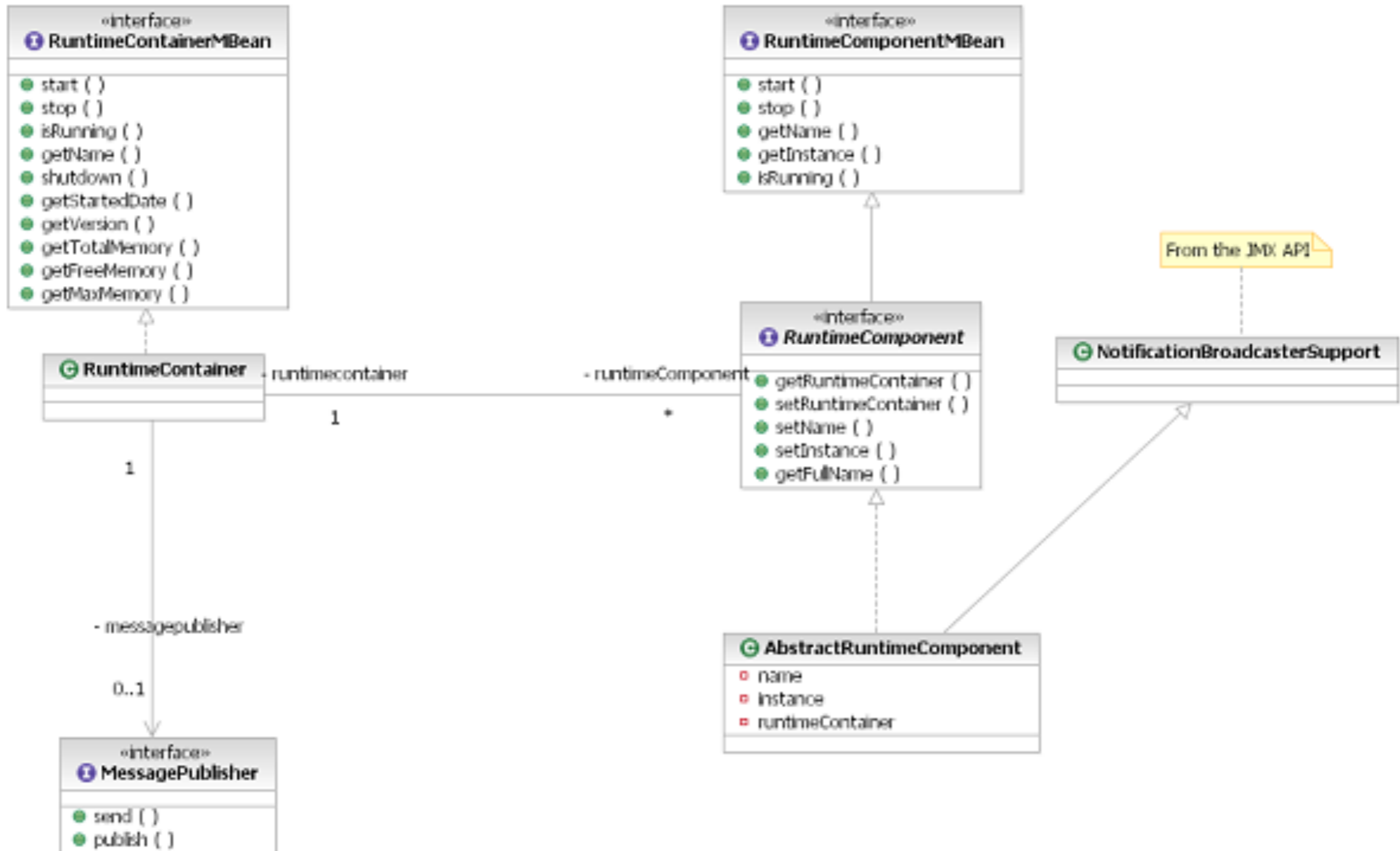
This is a Microsoft Visio block diagram that summarises the logical architecture of a financial trade life cycle manager (essentially an automated workflow system). It highlights the major interactions with external systems and the nature of those interactions (synchronous vs asynchronous).

Design View

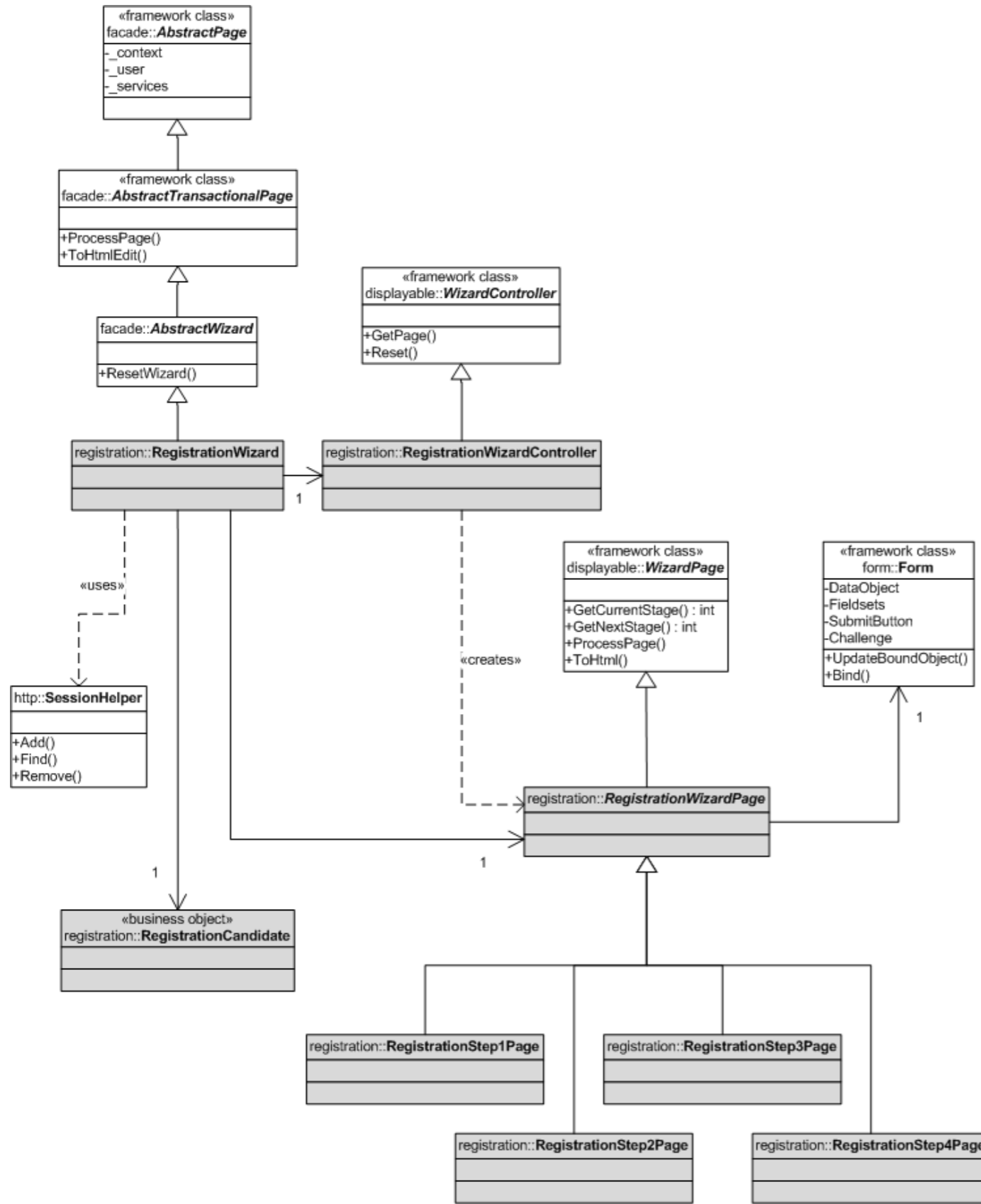
Design diagrams can provide a
summary of the important

implementation

details



This is a high-level UML class diagram that summarises the design for a bespoke runtime environment used to host application components. It shows the important interfaces along with their key operations and relationships.



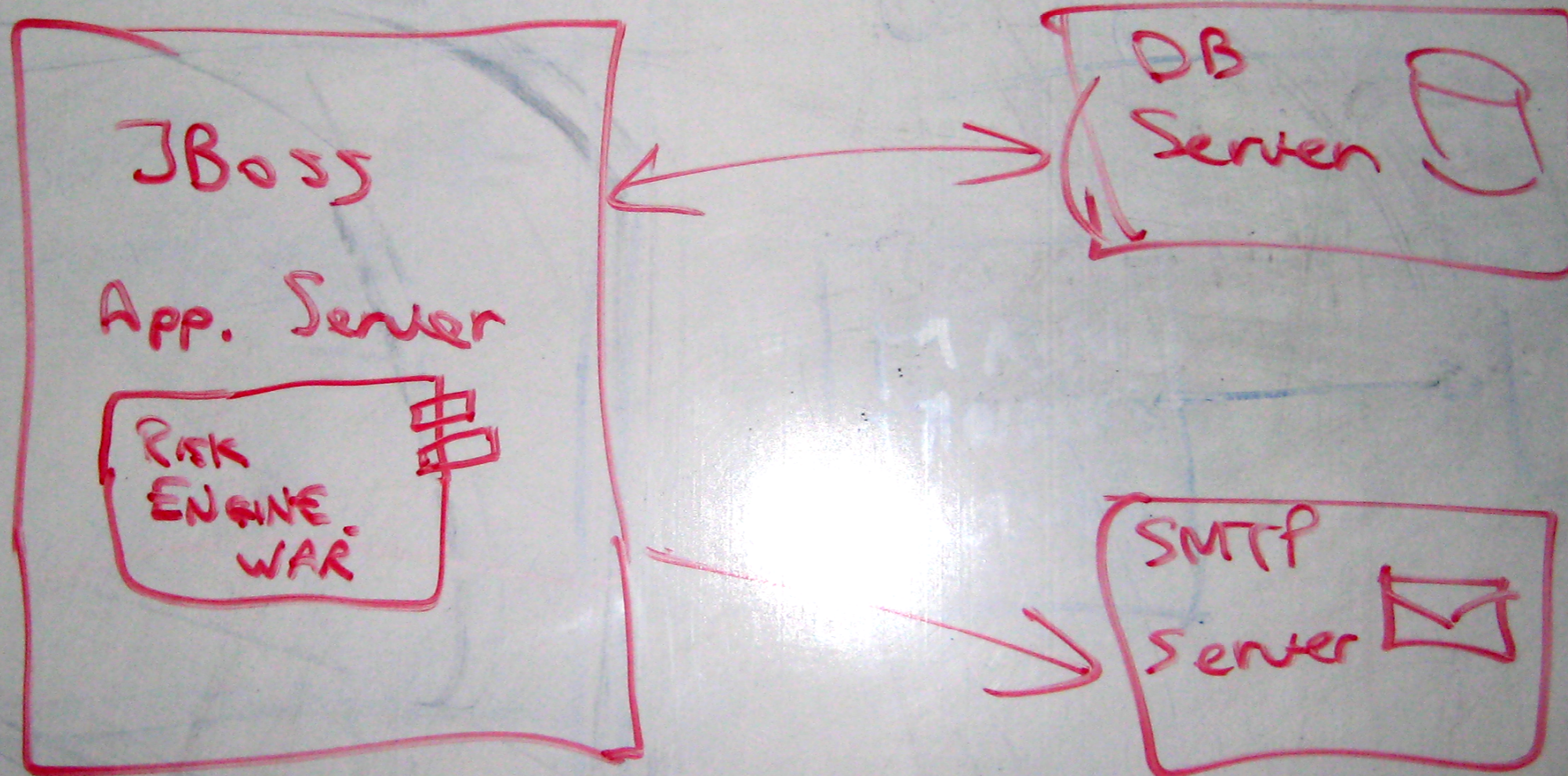
This is a high-level UML class diagram that summarises the design for some bespoke multi-web-page wizards. It shows the important classes along with their key properties and relationships.

A reference to a RegistrationCandidate instance is used throughout the classes shown on this diagram, but the associations are not shown for brevity.

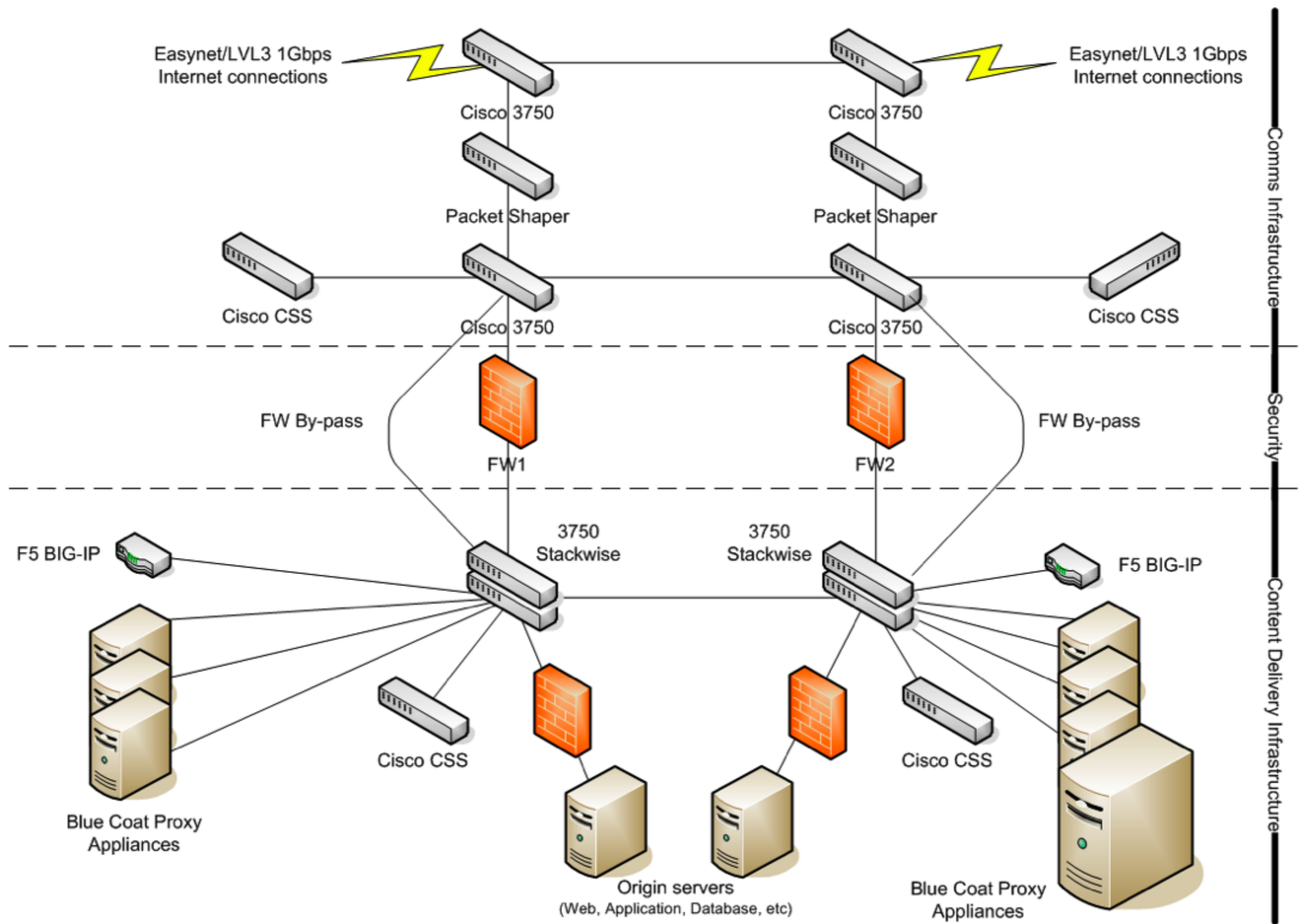
Infrastructure View

Infrastructure diagrams show the
physical hardware
and **infrastructure**
on which the software
will be deployed

INFRASTRUCTURE (+ DEPLOYMENT)

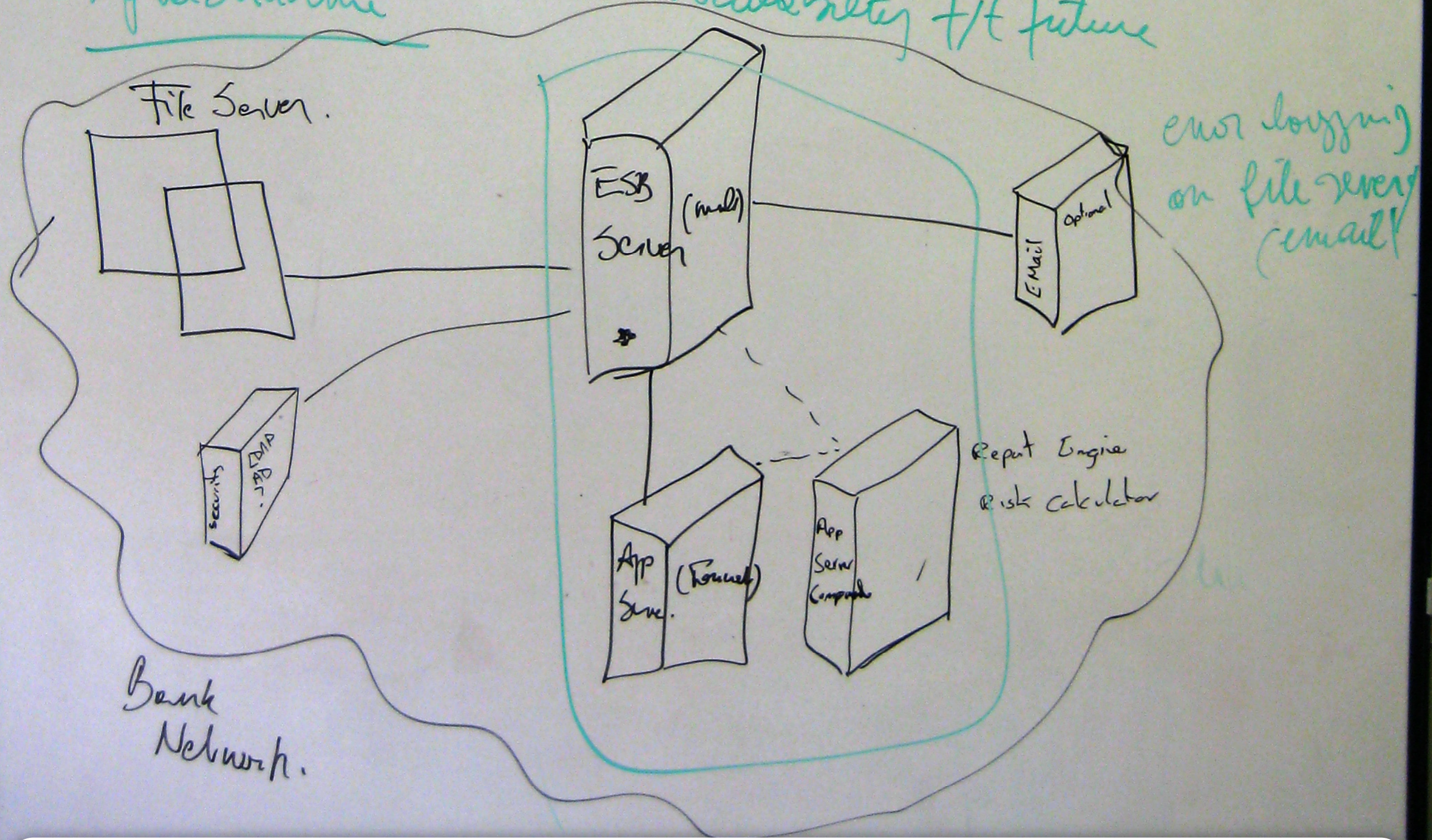


This diagram shows an overview of the different servers that make up a financial risk system.



This is a high-level Microsoft Visio network diagram that summarises the design for a large public facing website. It shows the important infrastructure components and their connections.

Infrastructure without ← = now / initial deployment
= scalability + / future



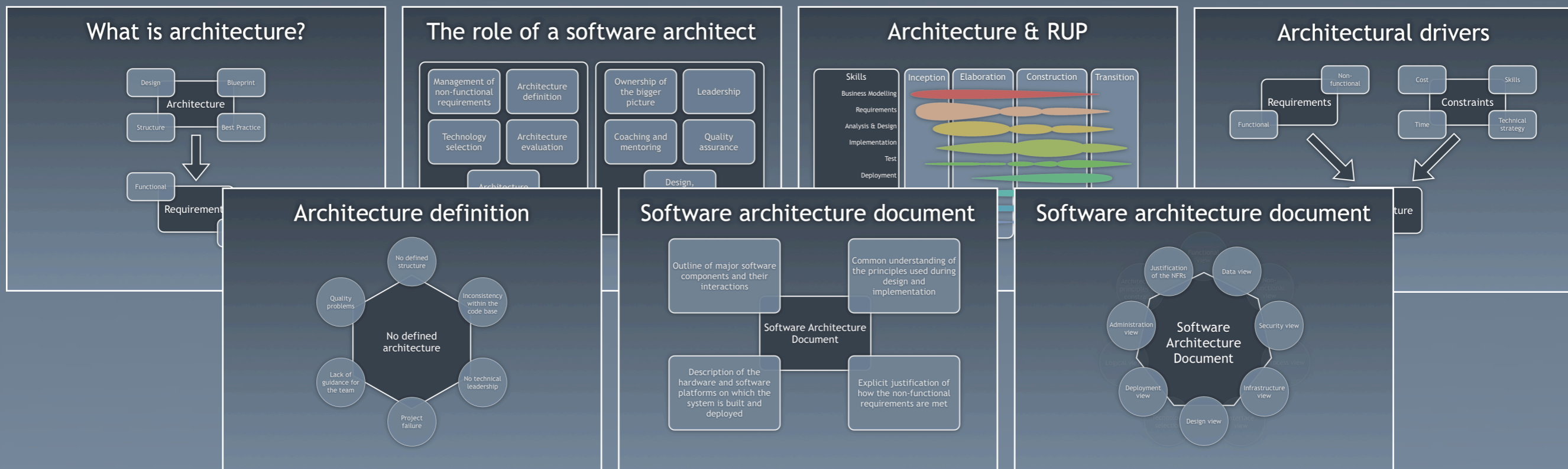
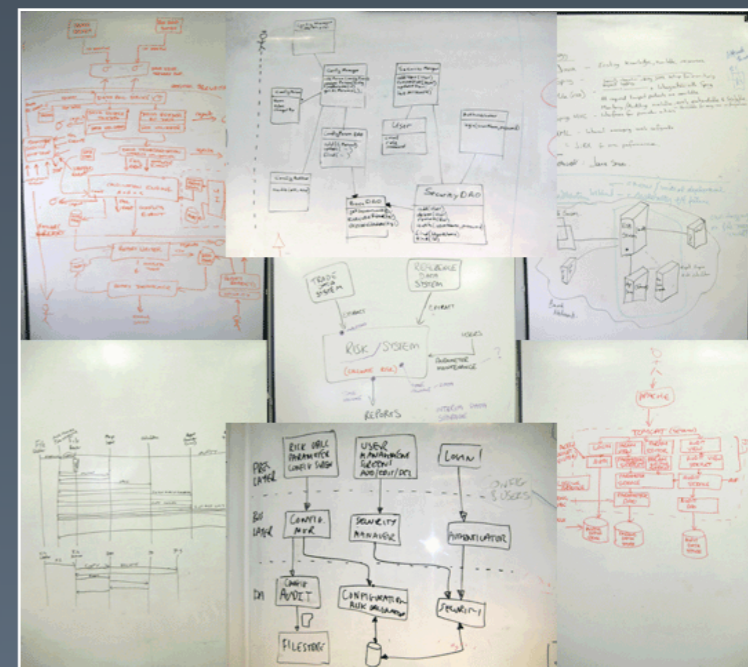
This diagram shows the major logical infrastructure nodes for a financial risk system.



From Developer to Architect

A 2-day interactive introduction to software architecture

Our “From Developer to Architect” training course is about broadening your software development skills and has been designed to take full advantage of the technical knowledge that you already have; whether that’s .NET, Java or something else. This course will make you more “architecturally aware”, helping you build better software. It’s about pragmatic and real-world software architecture rather than academic “ivory tower” software architecture.



coding {the} architecture



"Coding the Architecture" is a website and community for hands-on, pragmatic software architects. You'll find content and discussion about architecture and the role of an architect, along with our experiences of undertaking that role.

E-mail: simon.brown@codingthearchitecture.com

Website: <http://www.codingthearchitecture.com>

Google Group: <http://groups.google.com/codingthearchitecture>