

A Thinking Framework: from behaviour to structure.

Bavo De Ridder

Solution & Enterprise Architect

De Ridder Consulting

<http://www.bavoderidder.com/>



This work is copyrighted by De Ridder Consulting b.v.b.a. You are not allowed to alter, distribute, copy or transmit this work nor can you use this work for commercial or non-commercial purposes.

Some or all of the above restrictions can be waived with a written permission from De Ridder Consulting b.v.b.a.

For more information or enquiries, please contact via:

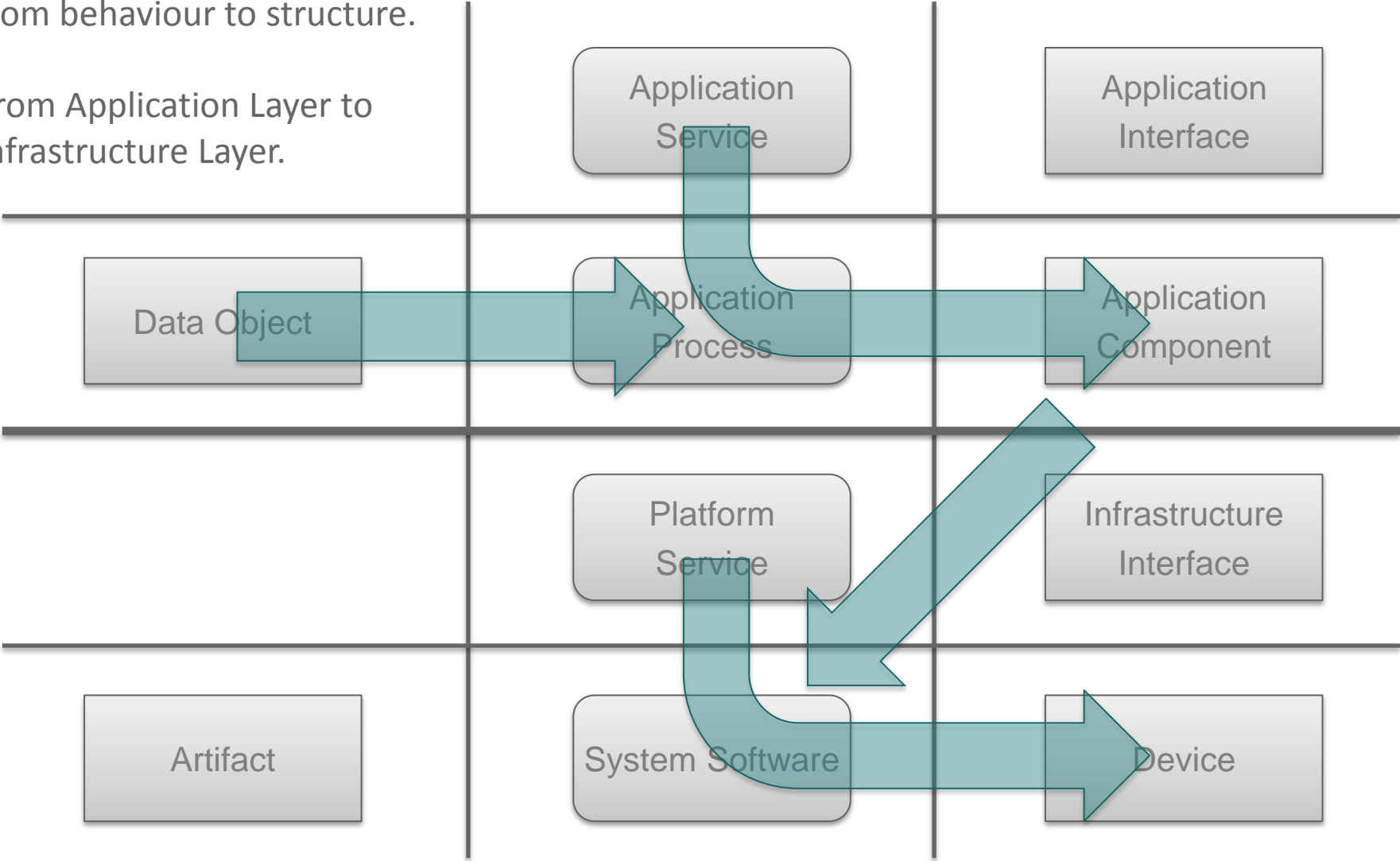
<http://www.bavoderidder.com/landing/contact>

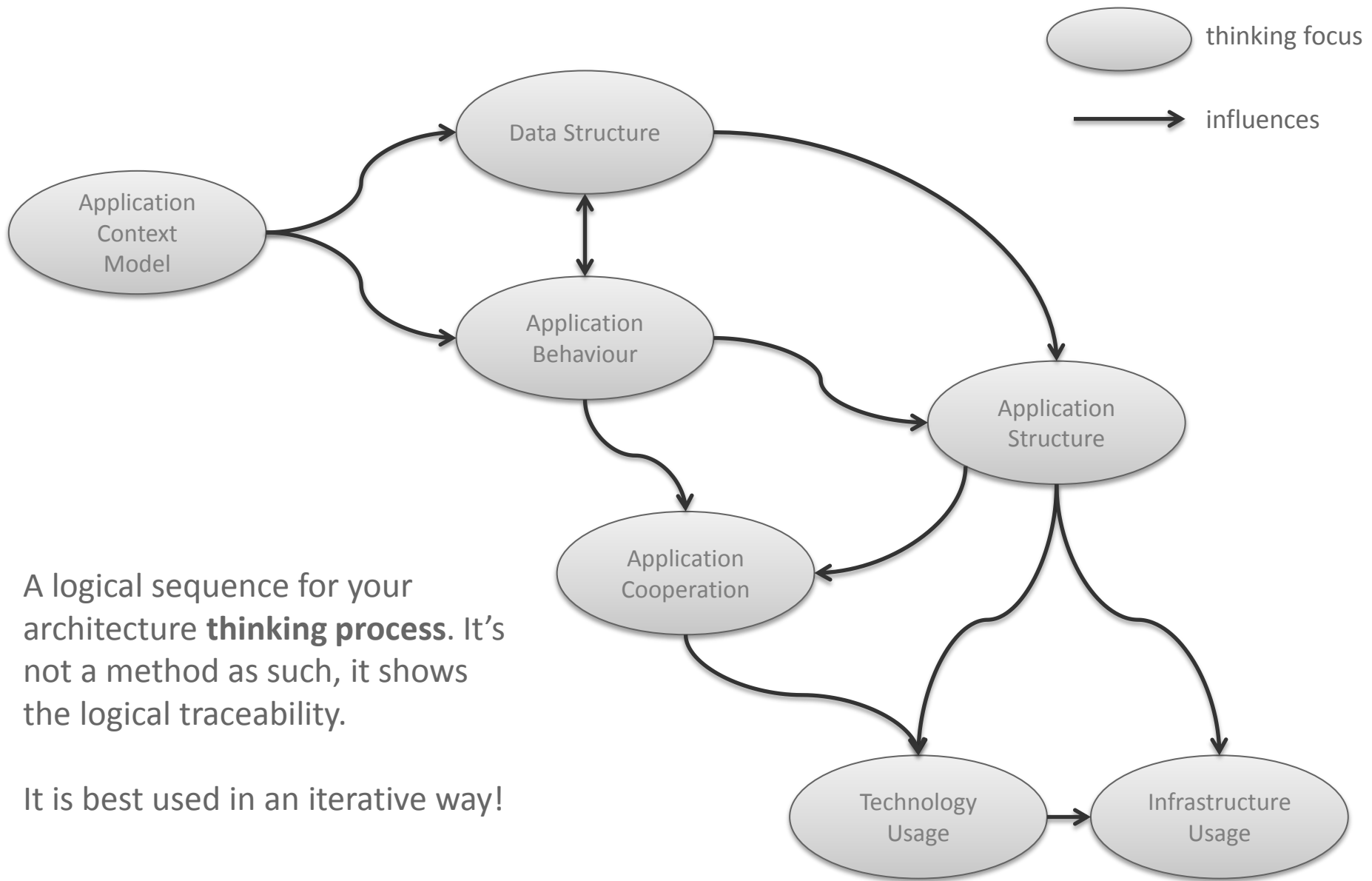
Thinking Framework

- There is a logical sequence to “construction”
 - External before internal
 - Behaviour before structure
- No matter what method you use, this logical sequence has to be a part of it.
 - It’s the logical traceability you can’t ignore.
- But ... it’s not a method to follow rigorously
 - Reality is iterative, insights gained and decisions made, will make you revisit.
 - There are other processes to take into account.

From external to internal,
from behaviour to structure.

From Application Layer to
Infrastructure Layer.





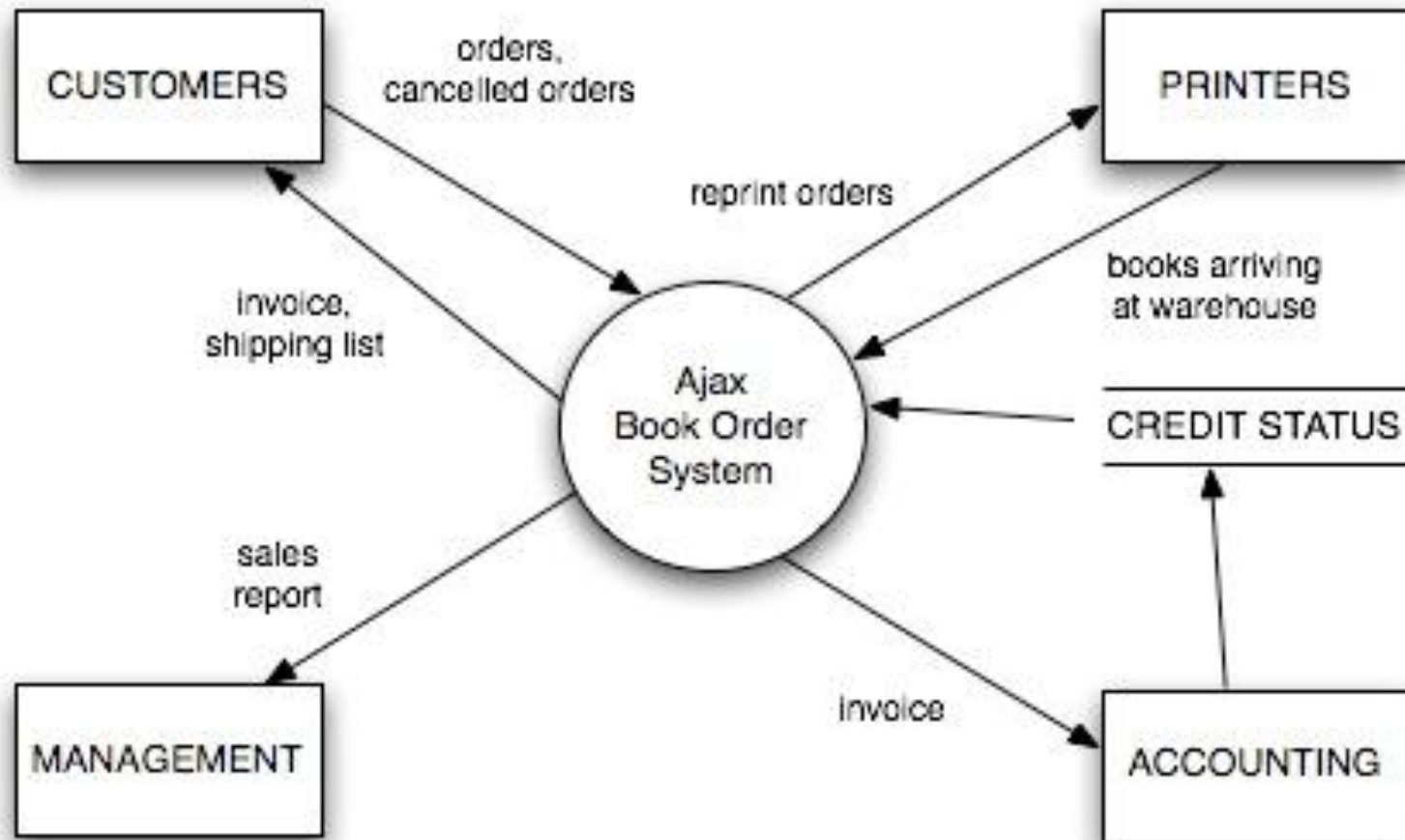
A logical sequence for your architecture **thinking process**. It's not a method as such, it shows the logical traceability.

It is best used in an iterative way!

Application Context Model

- A context model showing the boundary of the information system.
 - See also Volere (<http://www.volere.co.uk/>)
- Focus of thinking process
 - External actors
 - Other information systems
 - Human actors
 - Message flows to and from the system

Application Context Model

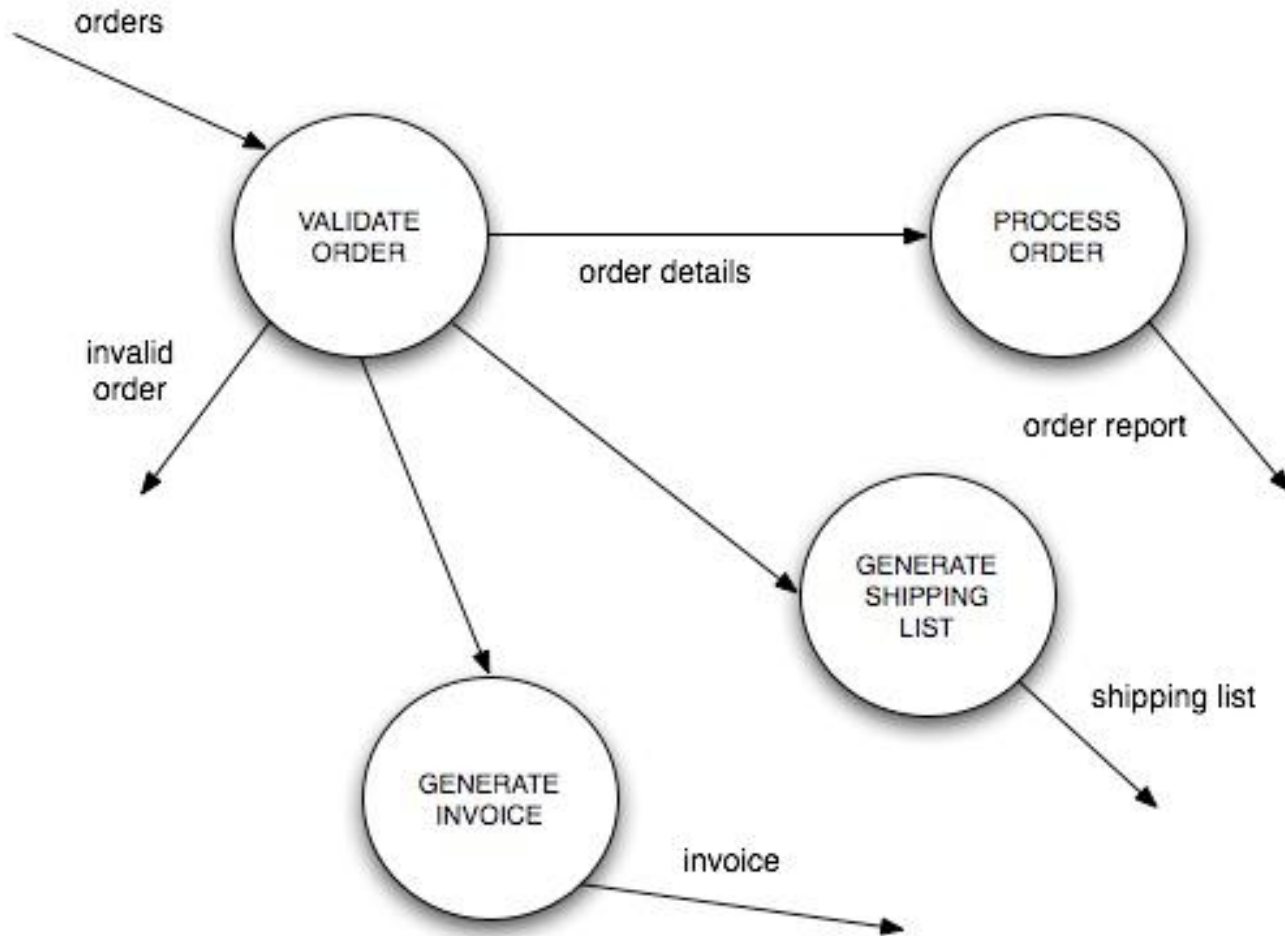


Application Behaviour

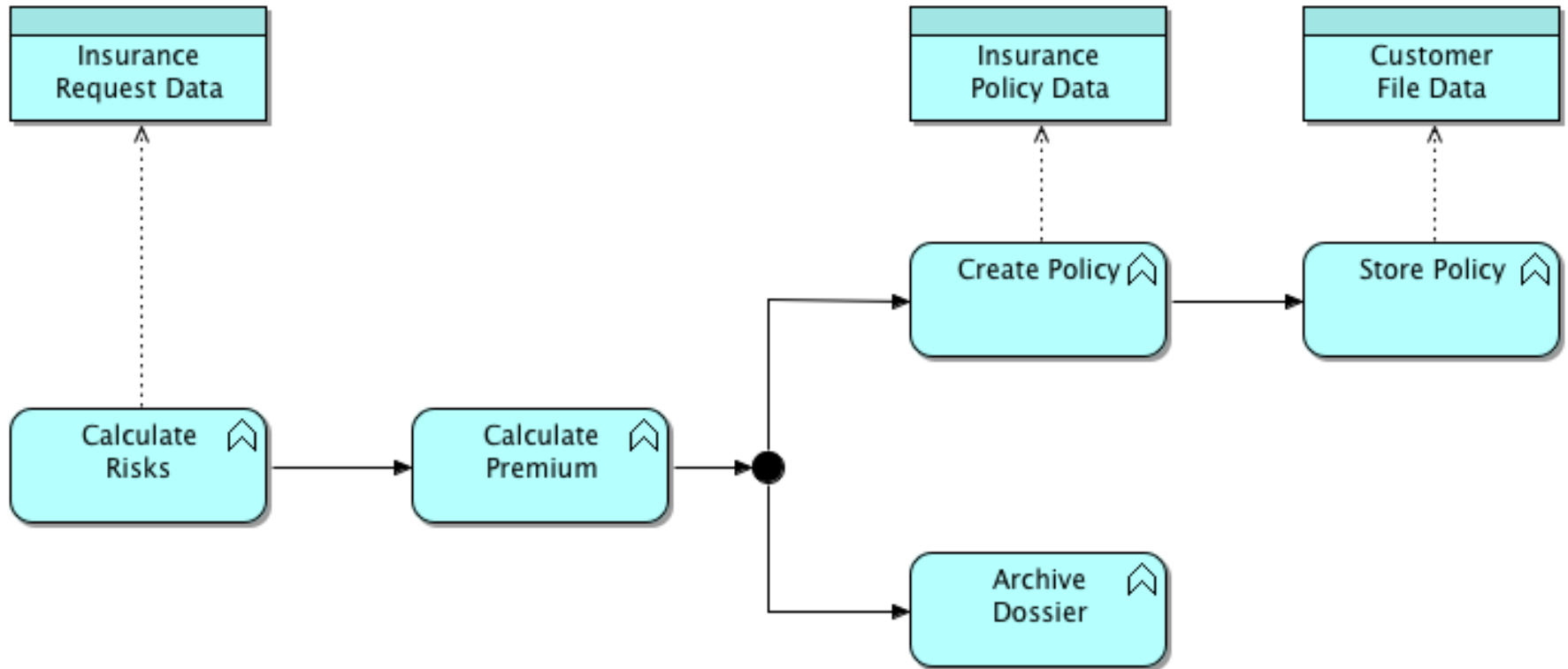
- Functional decomposition of the context model
 - See also DFD decomposition
- Focus of thinking process
 - Discover the behaviour of the future components
 - Discover the relations between the behaviour (*)
 - Discover logical repositories for data “in rest”
 - Postpone explicit structural thinking

(*) we wrap the behaviour in a component to show the inter-relationships, these are not necessarily the final components of the solution.

Application Behaviour



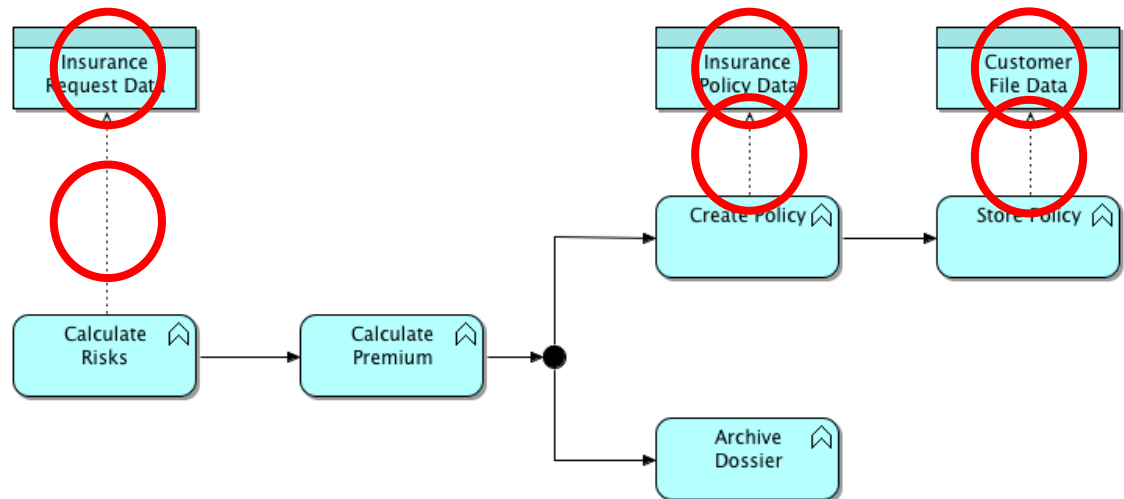
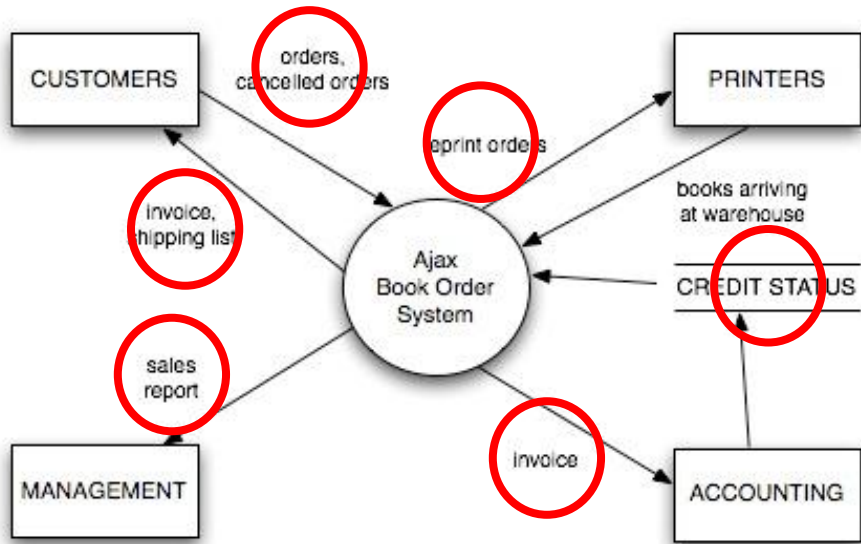
Application Behaviour



Data Structure

- Data structure of both data “in rest” (repositories) and data “in motion” (message flows)
- Focus of thinking process
 - Develop more insight in the data of the information system
 - Validate the previously identified behaviour (correctness and completeness), adjust where needed
 - Entity life cycle models as a way to enrich and verify the behaviour

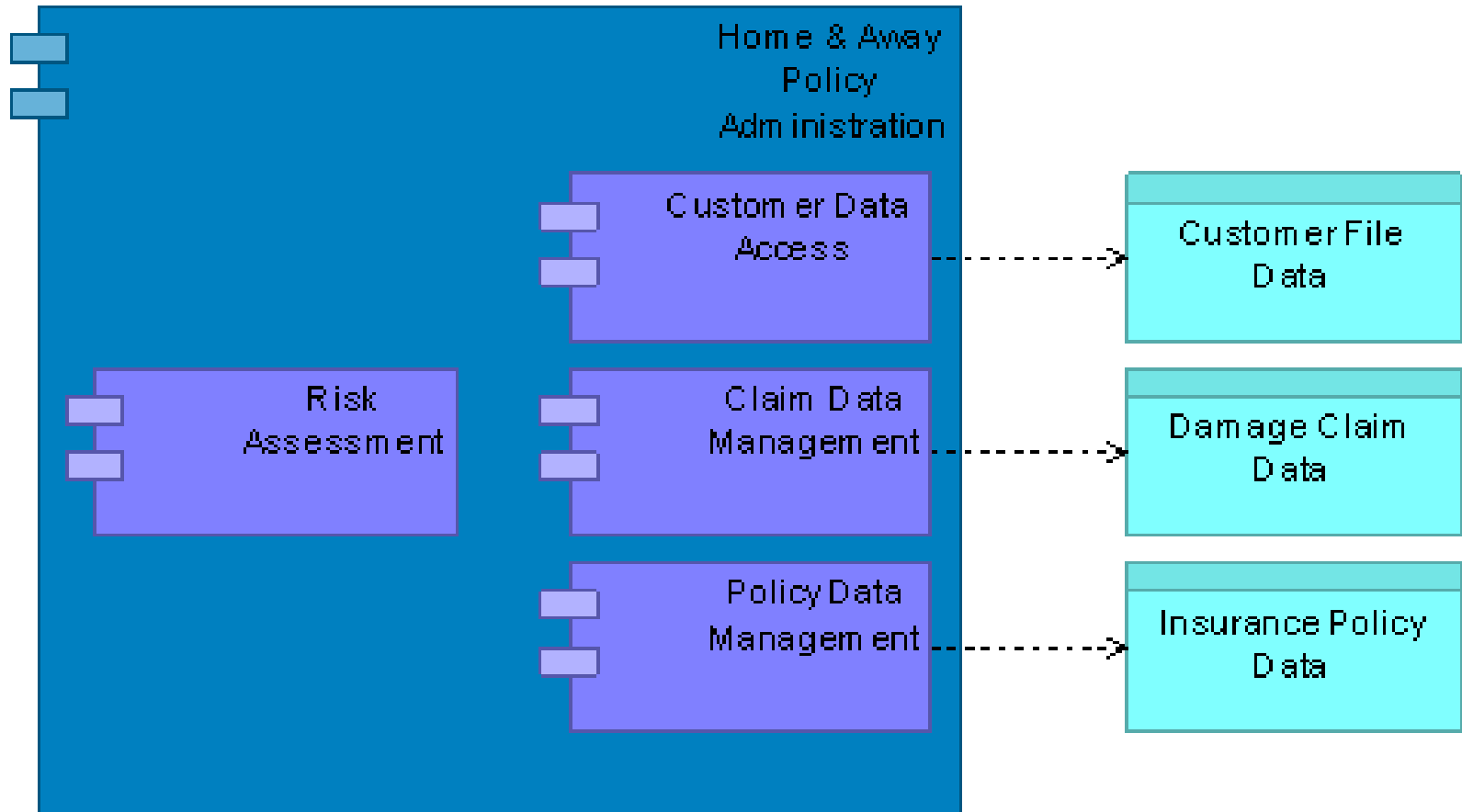
Data Is Everywhere



Application Structure

- A deliberately chosen structure to exhibit the required behaviour.
- Focus of thinking process
 - Analyse behaviour and the relationships with data
 - Find cohesive sets of behaviour for candidate components
 - Apply non-functional requirements, standards, principles and other constraints or directives.

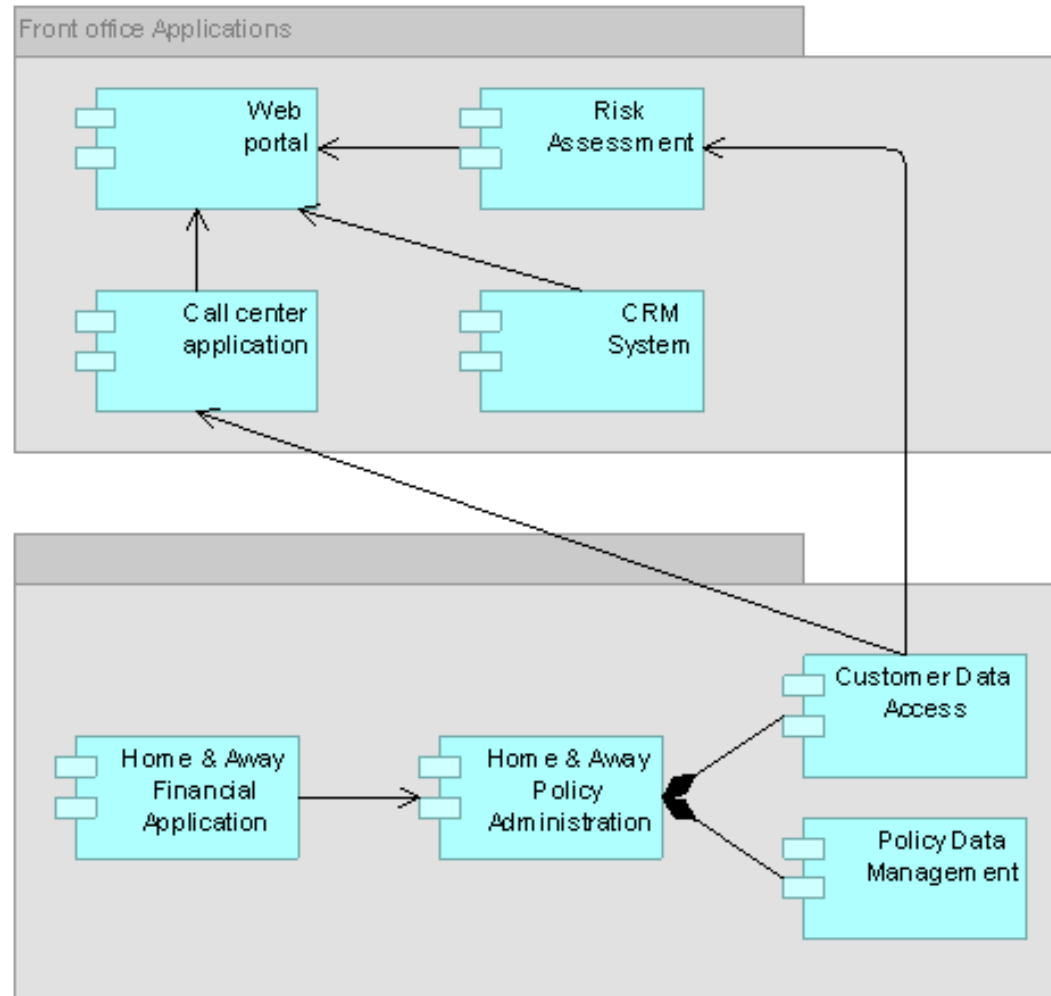
Application Structure



Application Cooperation

- Show the dependencies (uses/used by) between application components.
- Focus of thinking process
 - Transfer the behavioural relations to the application components exhibiting the behaviour.
 - Identify offered and required interfaces to enable the relations between applications

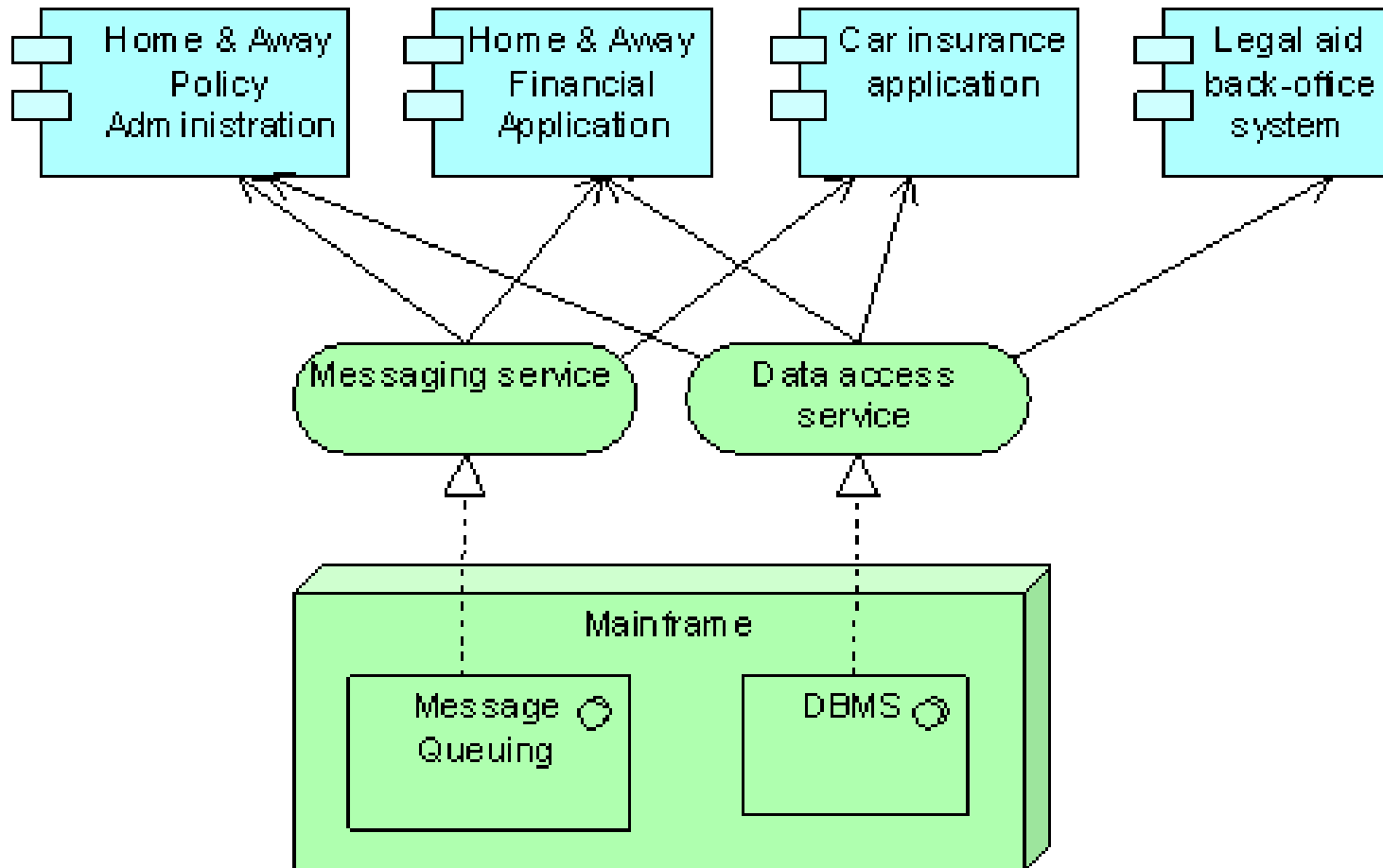
Application Cooperation



Technology Usage

- Show the technology used to realize the chosen application structure.
- Focus of thinking process
 - Maximization of standardisation benefits.
 - Realization with commonly available technology.

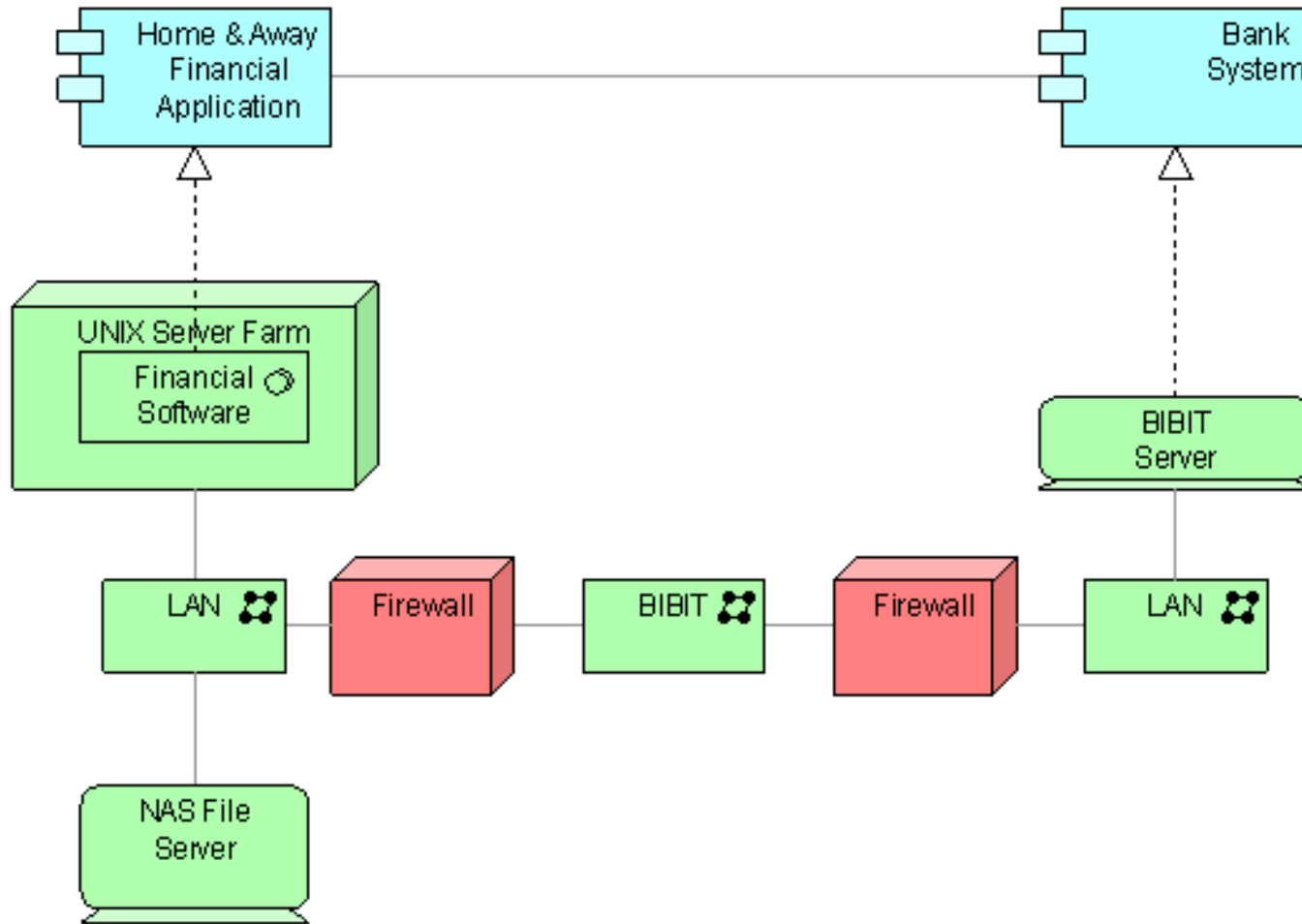
Technology Usage



Infrastructure Usage

- Shows the used hardware and the actual locations where applications and technology will be housed.
- Focus of thinking process
 - Physical location for application hosting.
 - Networks involved to connect all elements.
 - Hardware required (real and virtual).

Infrastructure Usage



Abstraction

- The “Realization Spiral”
 - Decisions made allow you to revisit the entire process and create a new, “less abstract”, version of the architecture.
- It is however not the same architecture.
 - Any changes in the decisions made, only render the less abstract version obsolete.

Documenting Architecture

- No matter the method used, the key steps in the thinking process have to be documented.
 - Any change, now or later, will directly influence specific steps and decisions made in the thinking process.
- What to document, for each step:
 - Which decisions have been made and why?
 - What input was used and what output was produced?
- What level of detail to use?
 - How significant was it in the thinking process?

Thank you!

Bavo De Ridder

Solution & Enterprise Architect

De Ridder Consulting

<http://www.bavoderidder.com/>