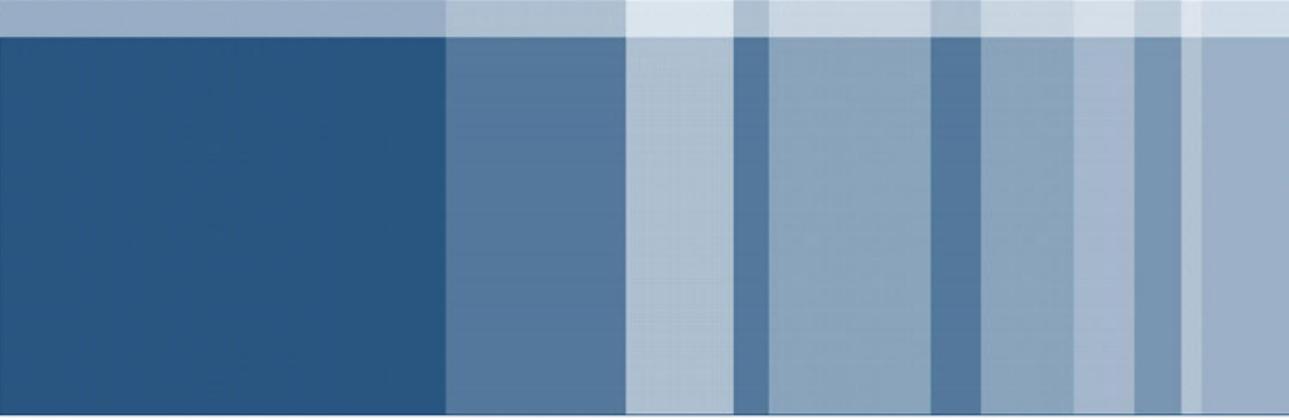




Sistemi Informativi (per il settore dell'informazione) – Prof. Barbara Pernici

22 novembre 2018

POLITECNICO DI MILANO



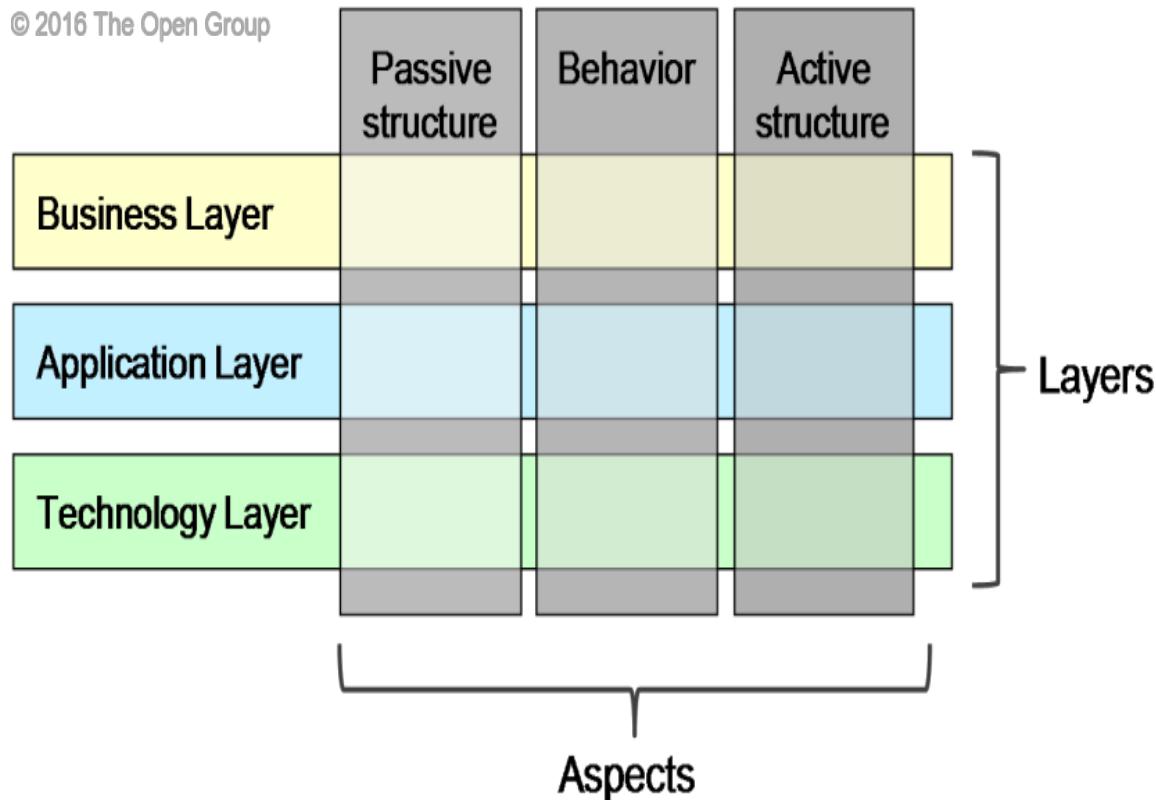
Gabriele Scalia

ARCHIMATE PARTE 1



Archimate core

© 2016 The Open Group

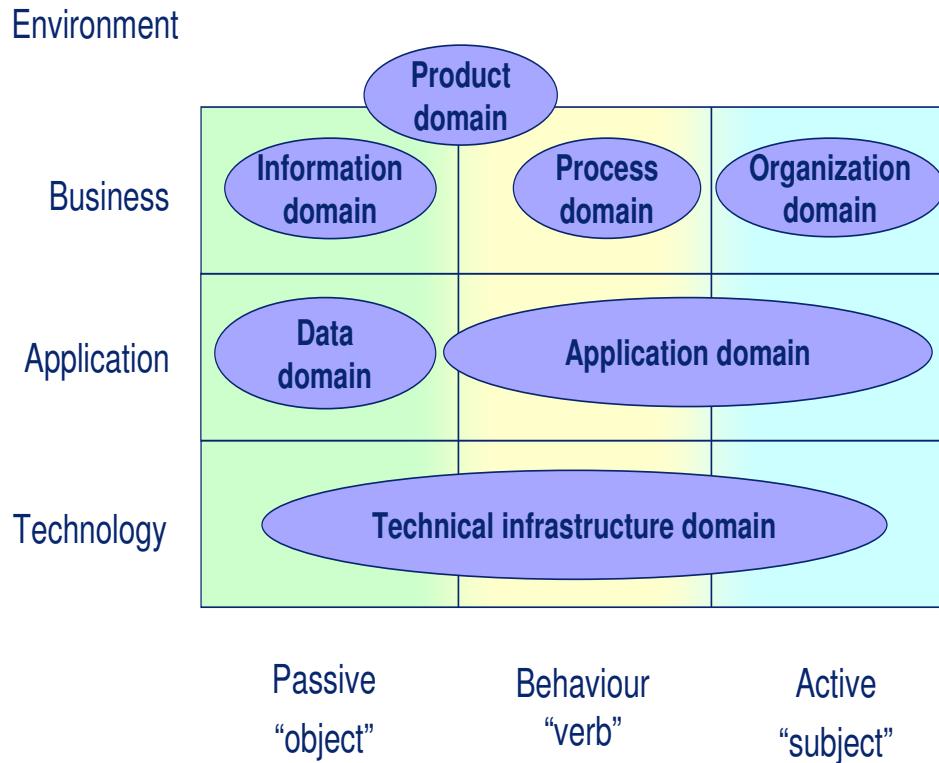




The components

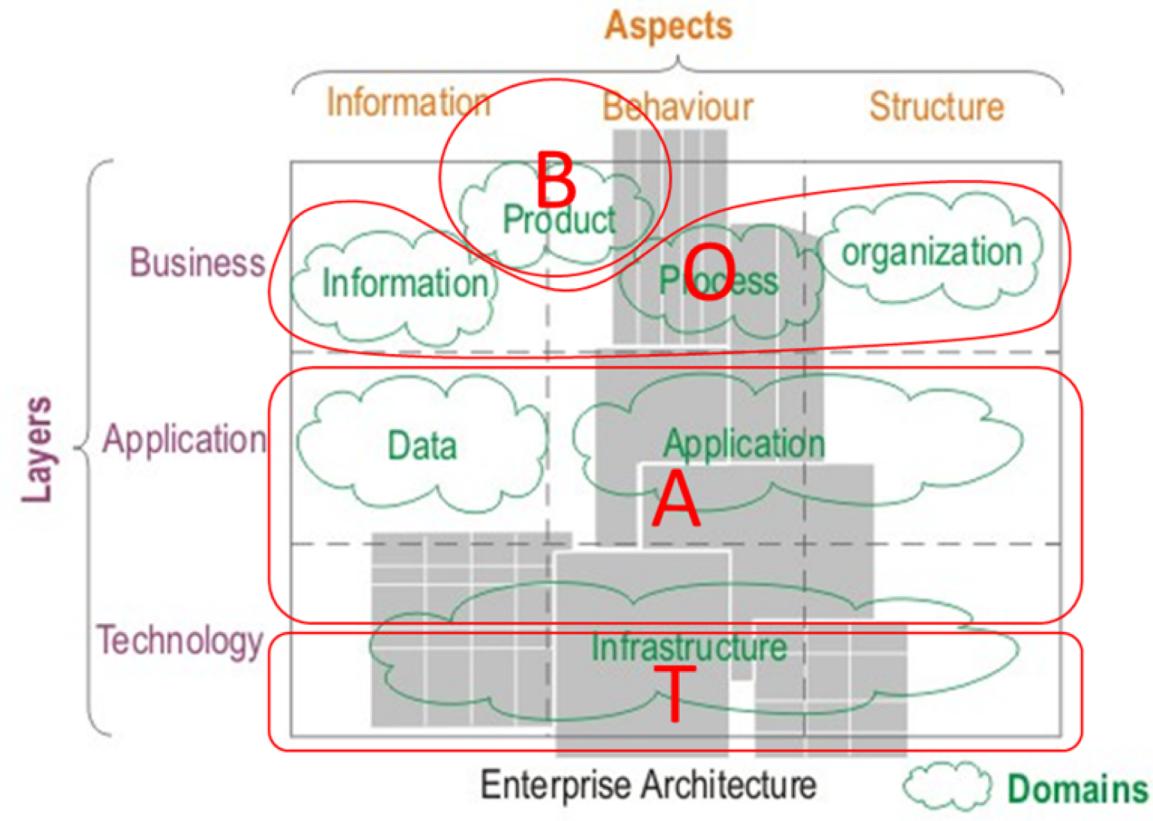
Three main components:

- **Active** elements (who)
- **Behavior** elements (acts)
- **Passive** elements (on)



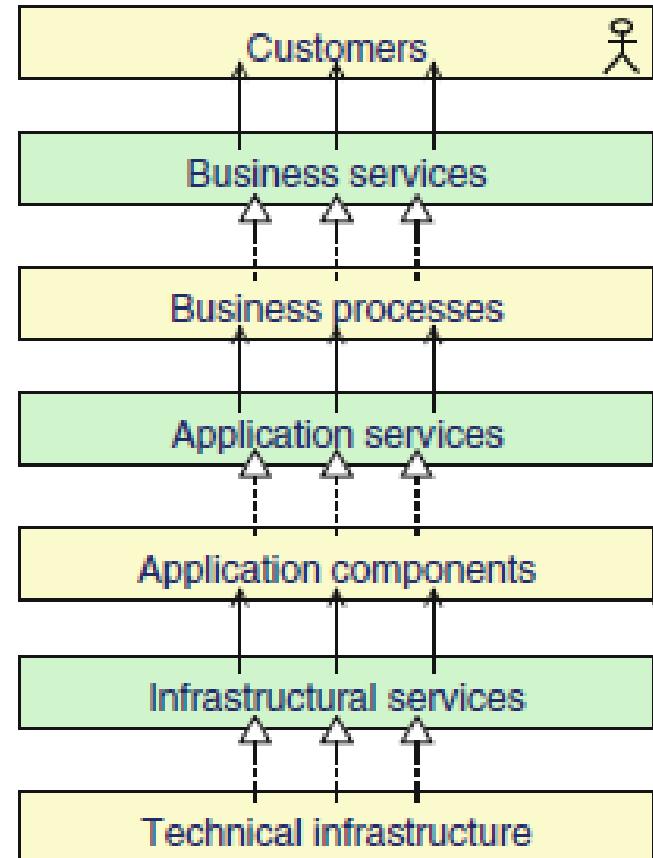


Archimate vs BOAT



Service Oriented Architecture

- Components (application, business, software) provide services to other components
 - Web services
- Software (SaaS), Platform (PaaS), Infrastructure (IaaS)
- **Services** are central to Archimate's architecture models



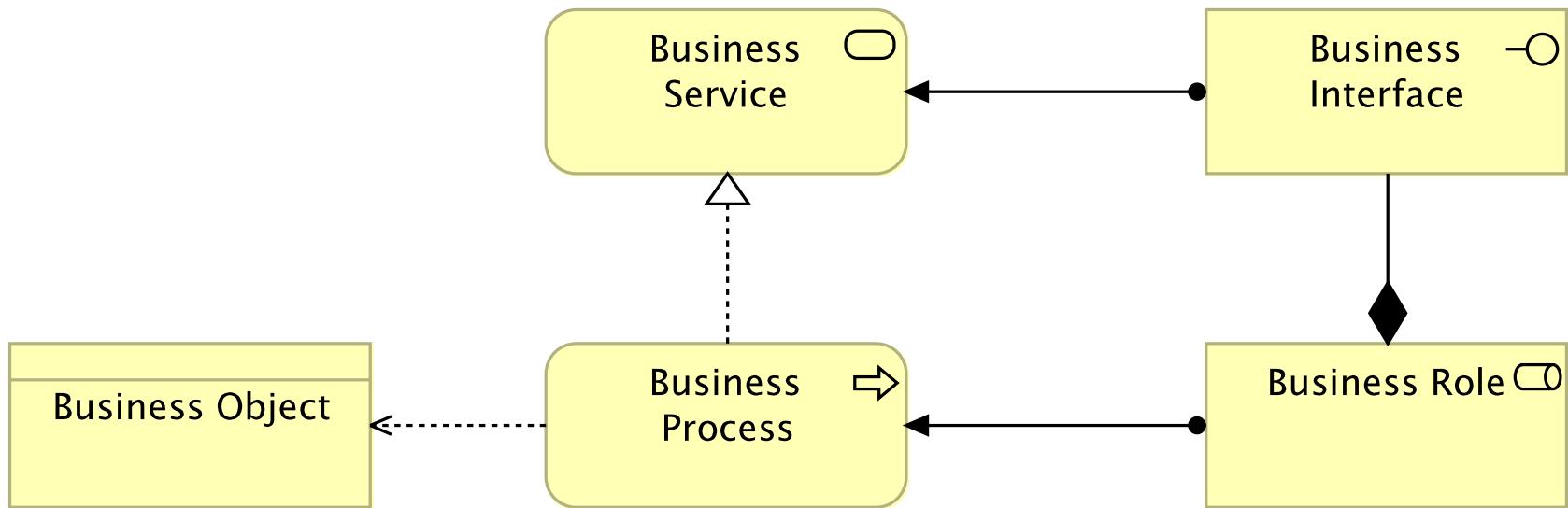


Archimate patterns

- In the following we see a basic “pattern” for each layer
- The pattern includes all the elements for a layer and their typical relationships
- In real cases most of the elements are not represented and relationships are derived (see later)
- In general, we include in a diagram only what we need to show



Basic business pattern



← This is the Access relation

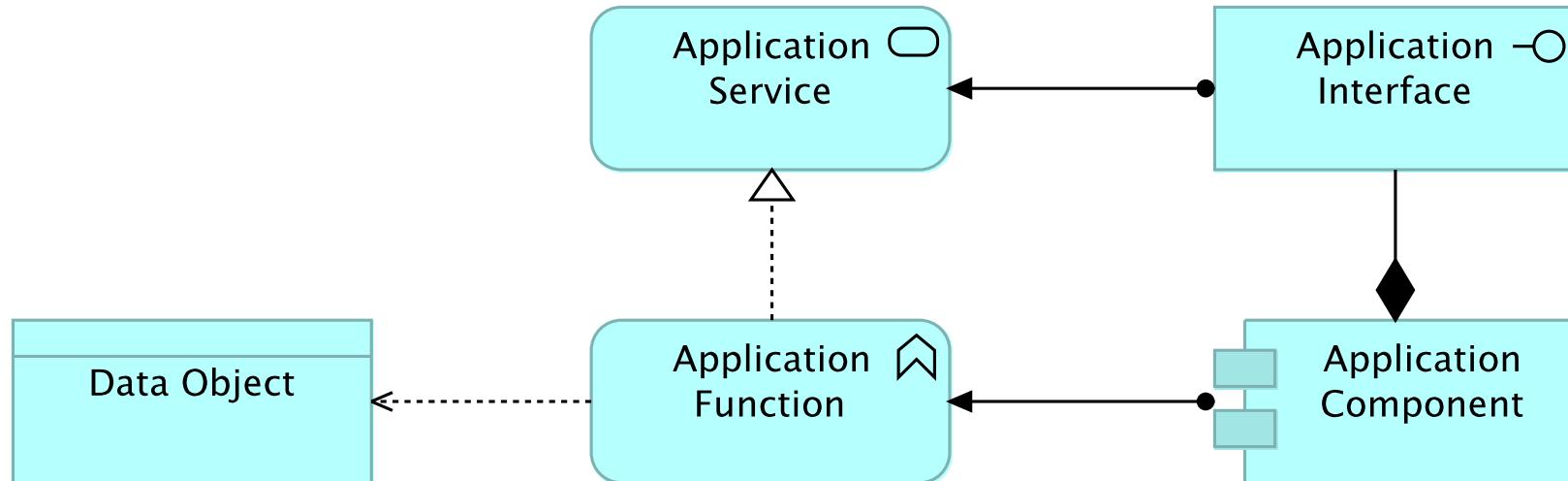
—————→ This is the Composition relation

—————→ This is the Realization relation

←————● This is the Assignment relation



Basic application pattern



← This is the Access relation

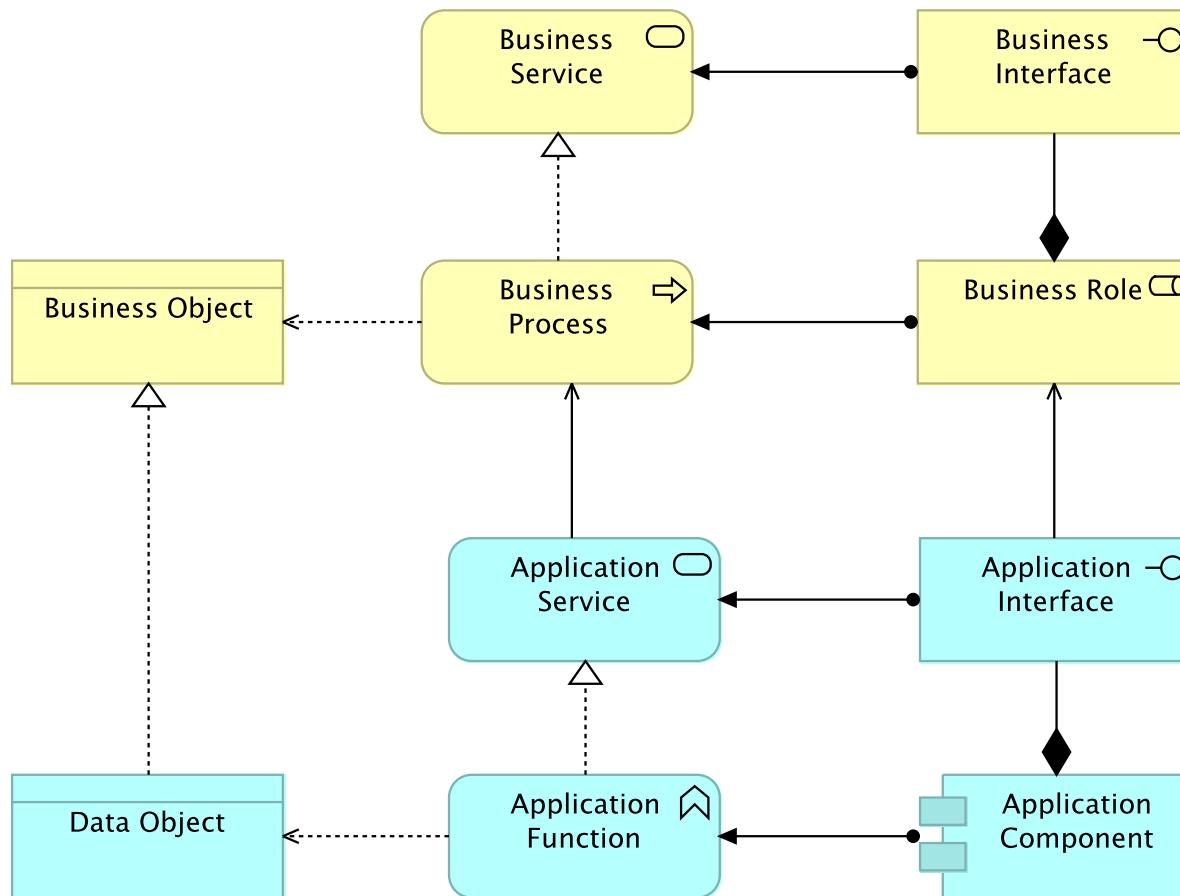
→ This is the Composition relation

↔ This is the Realization relation

↔ This is the Assignment relation



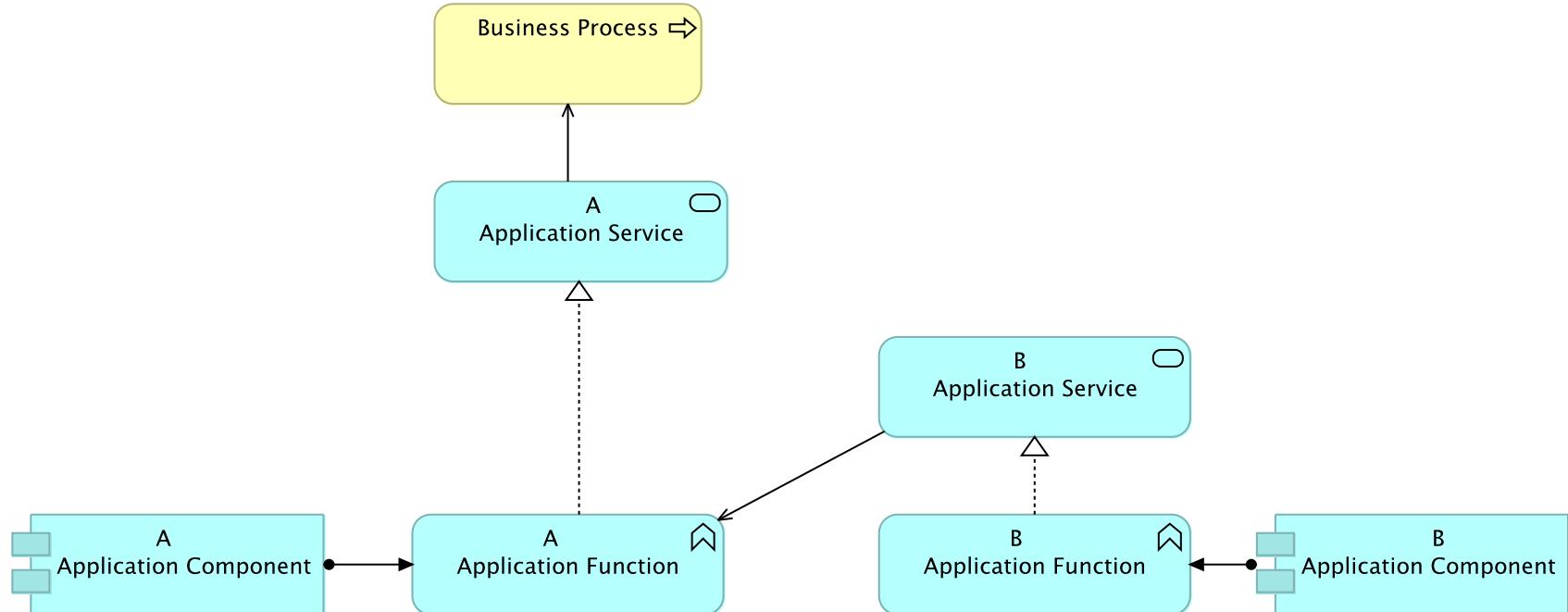
Basic application is used by basic business pattern



← This is the **Serving** relation



An application **is used by** another application

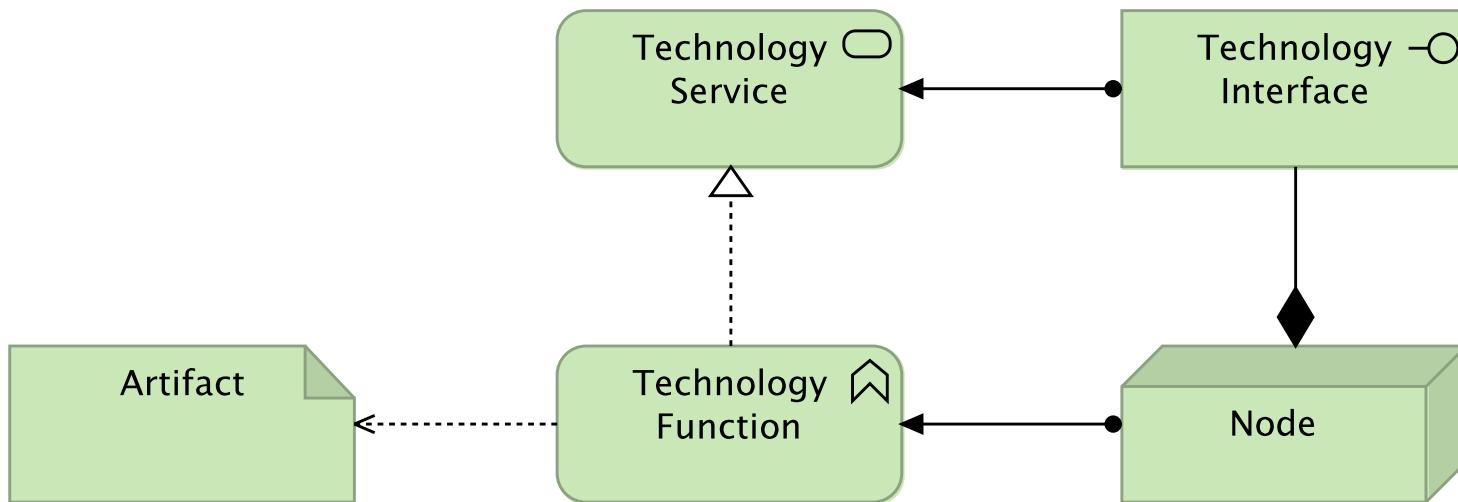


← This is the **Serving** relation

The serving relation can be used not only **between** levels of the architecture (previous slide), but also **within** a level (this slide)



Basic infrastructure



← This is the **Access** relation

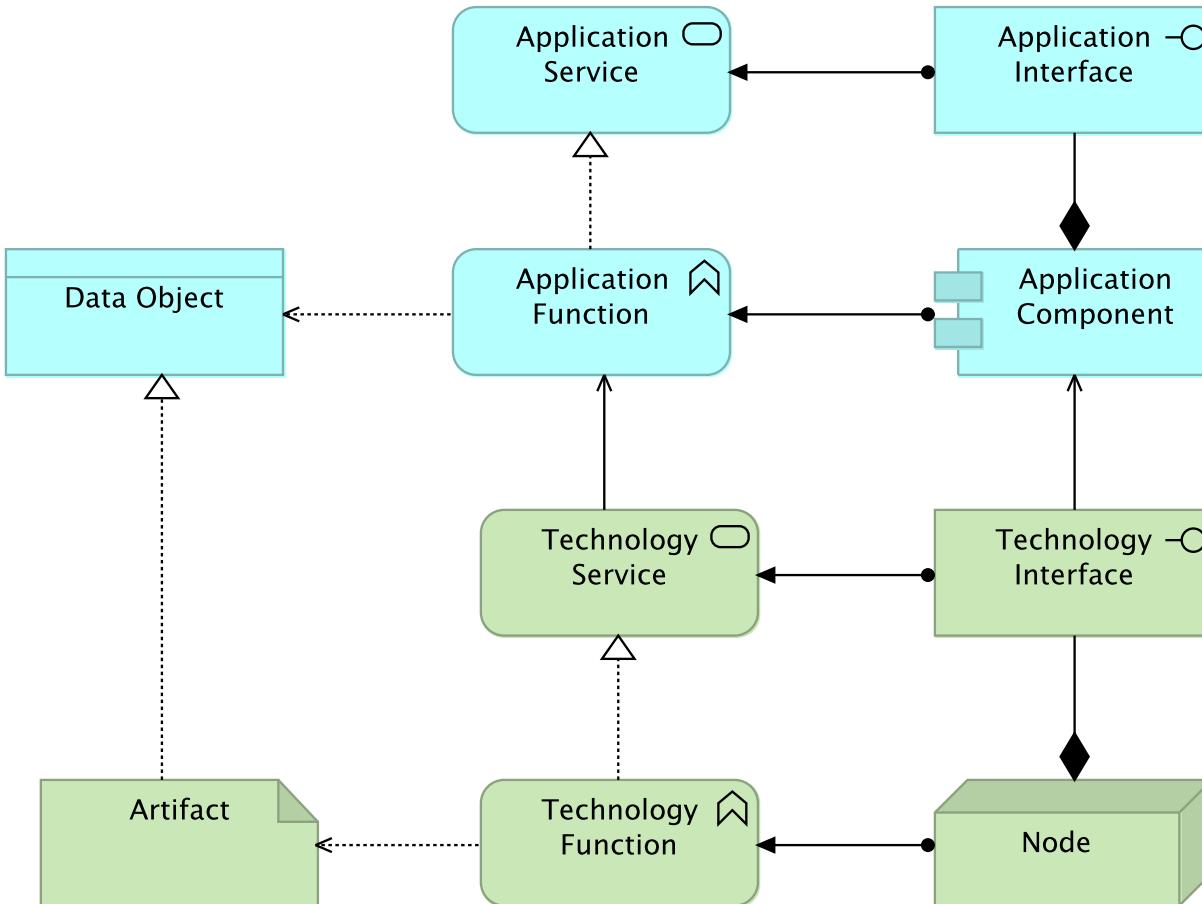
◆ This is the **Composition** relation

→ This is the **Realization** relation

↔ This is the **Assignment** relation



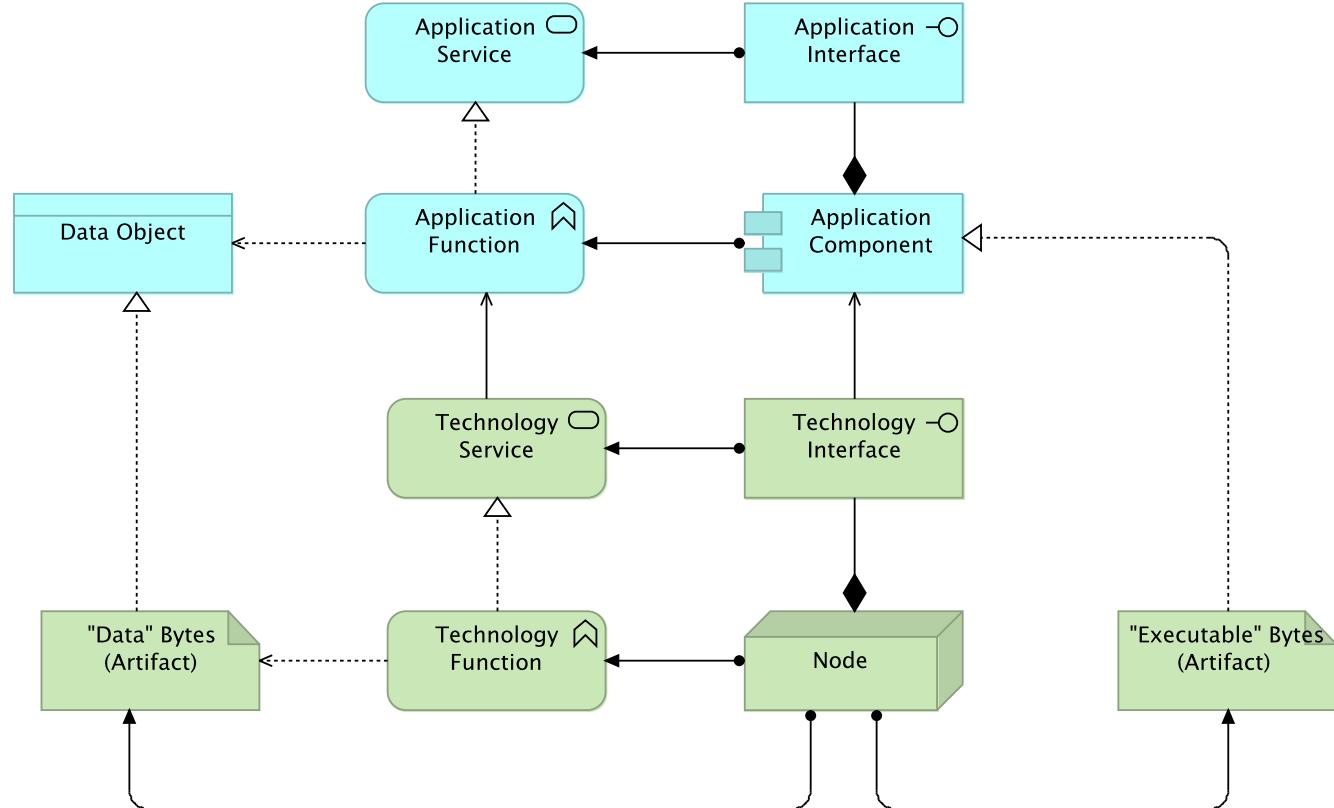
Basic application and basic infrastructure



← This is the **Serving relation**



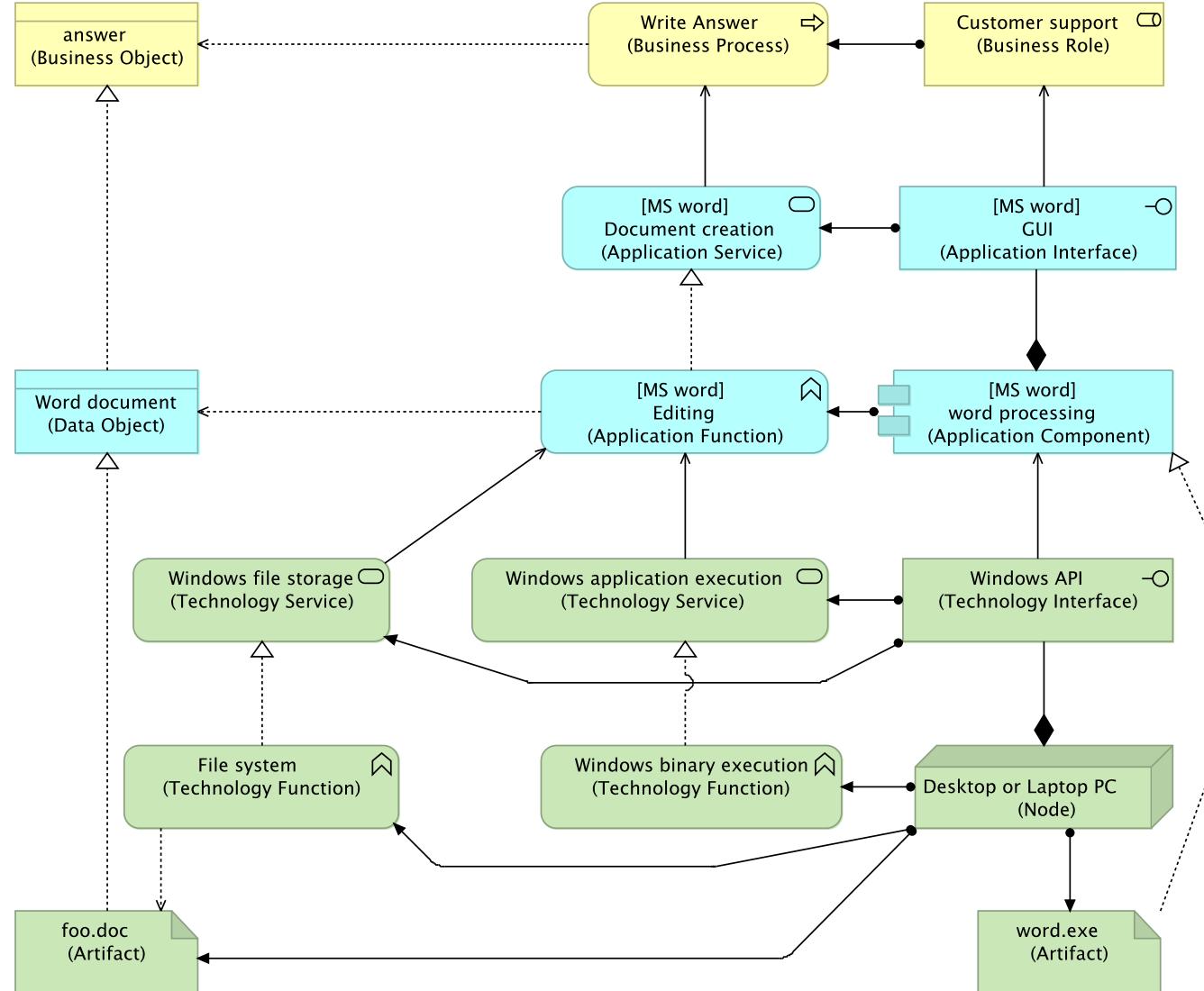
Basic application and basic infrastructure Extended



- The assignment relationship ($\leftarrow \bullet$) between Node and Artifacts stands for the fact that the Artifacts **resides** on the Node
- The “executable” artifact realizes ($\cdots \rightarrow \diamond$) the application



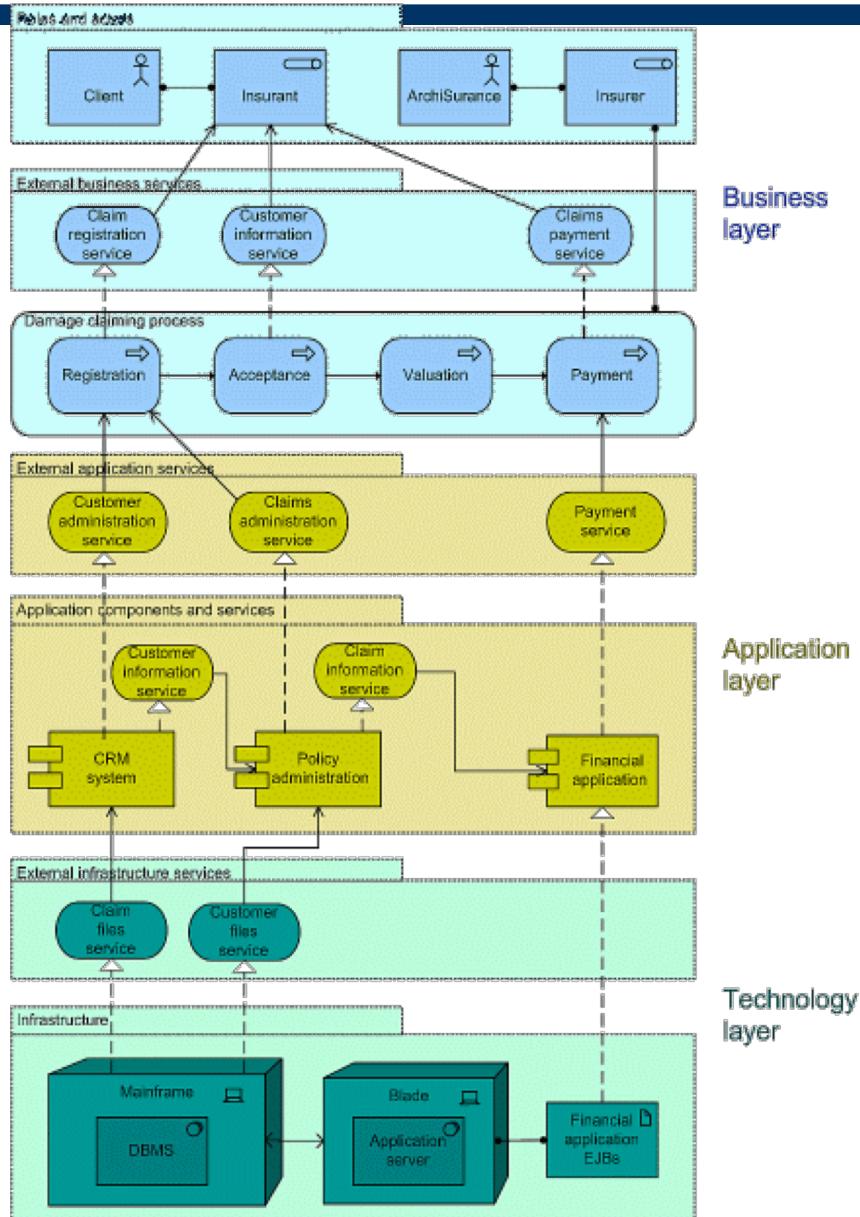
Example: putting all together



The model shows someone writing a letter to a customer that contains an answer, supposedly to a question the customer has asked. Two infrastructural services are needed for this to work. The application should run and the document must be stored. In this example, everything happens on a standalone PC.

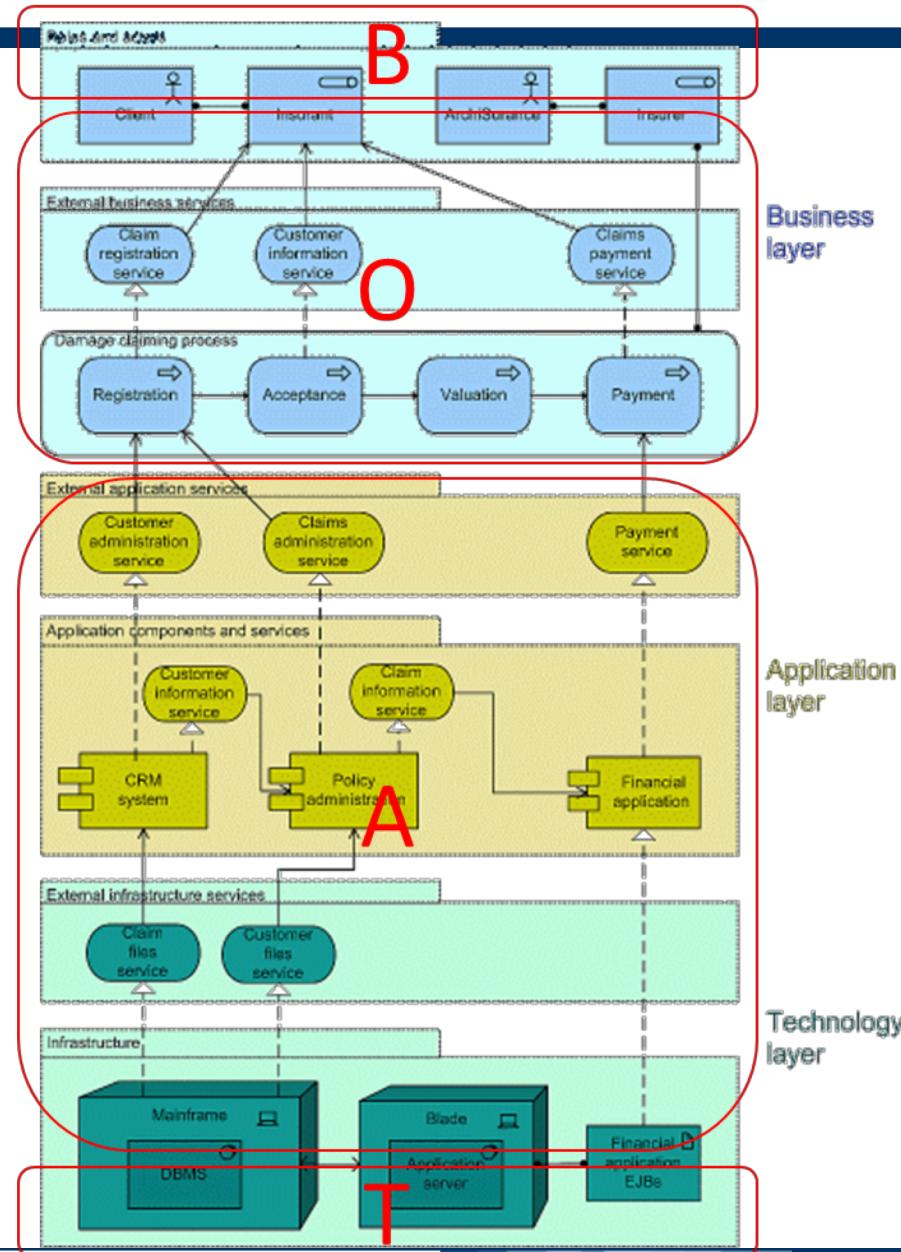


Archimate example





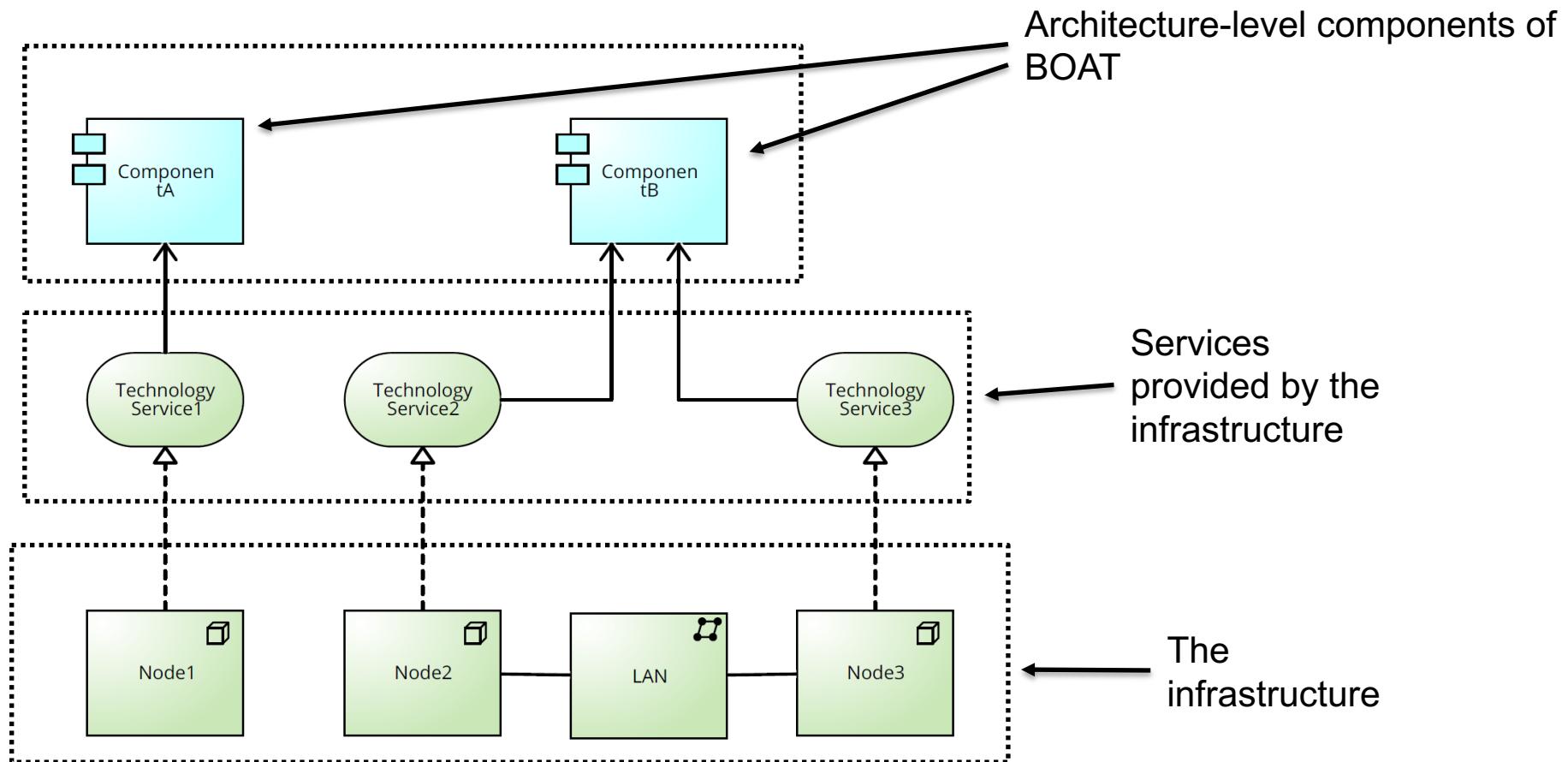
Archimate vs BOAT





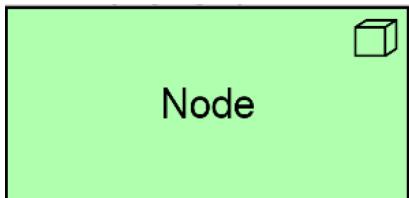
Modeling the infrastructure

- Our perspective: modelling the infrastructure as **service** for application components





Infrastructure: details



- A computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources



- A collection of technology behavior that can be performed by a node



- An explicitly defined exposed technology behavior

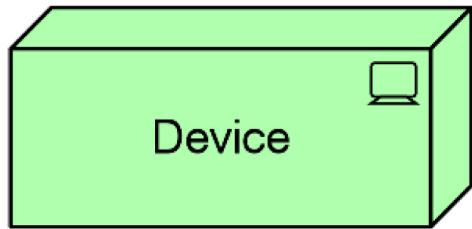


- A point of access where technology services offered by a node can be accessed

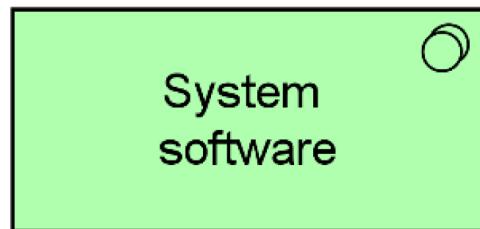
- Technology function and technology interface will be omitted in the following



Infrastructure: details



- A physical IT resource upon which system software and artifacts may be stored or deployed for execution
This is a purely hardware node



- Software that provides or contributes to an environment for storing, executing, and using software or data deployed within it
This is a purely software node



Path and network

Communication
Network

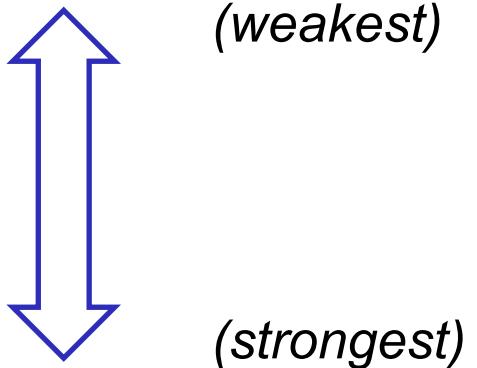


- A set of structures and behaviors that connects computer systems or other electronic devices for transmission, routing and reception of data or data-based communications such as voice and video
- **Association relation:** it connects nodes and networks



Derived relations

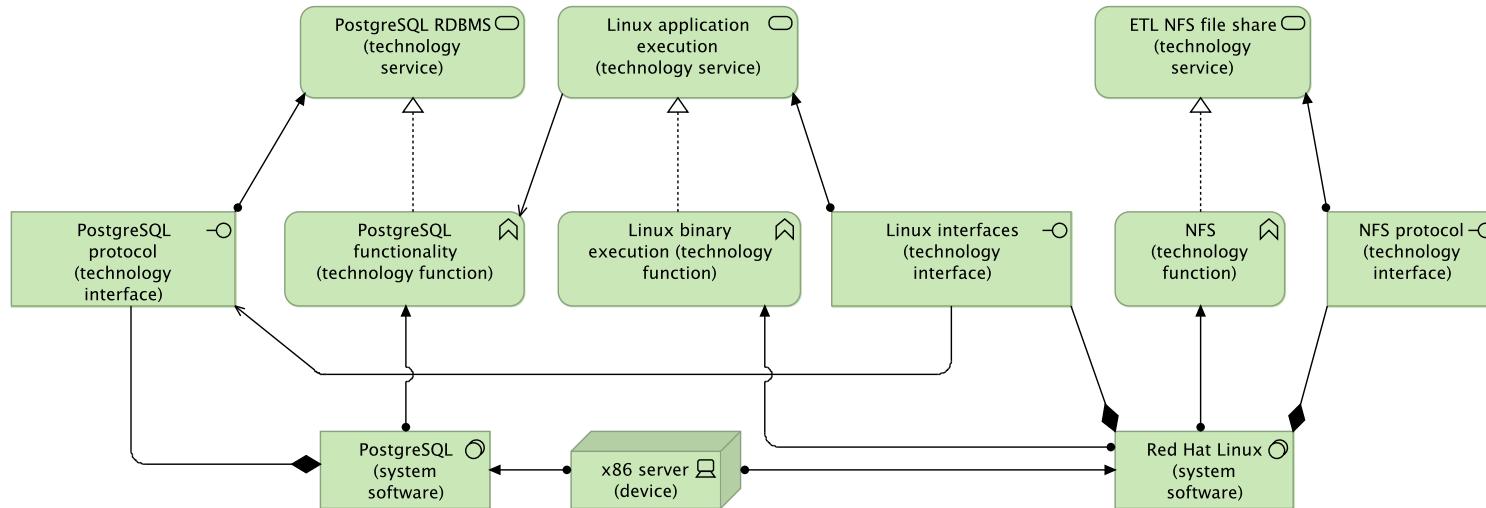
- Structural and dependency relations ordered by “strength”:
 - Access
 - Serving
 - Realization
 - Assignment
 - Aggregation
 - Composition
- Rule: two relationships that join an intermediate element can be combined and replaced by the weaker of the two
- Transitively applying this property allows to replace a chain of relationships with the weakest of the chain
- The resulting derived relationships are valid relationships in Archimate



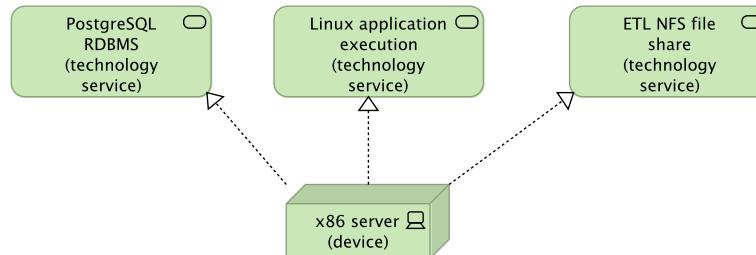


Derived relations: example

- “Standard” relations



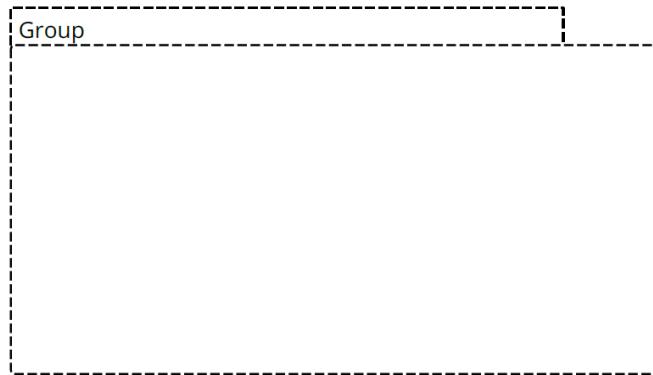
- Same model – derived relations





Grouping

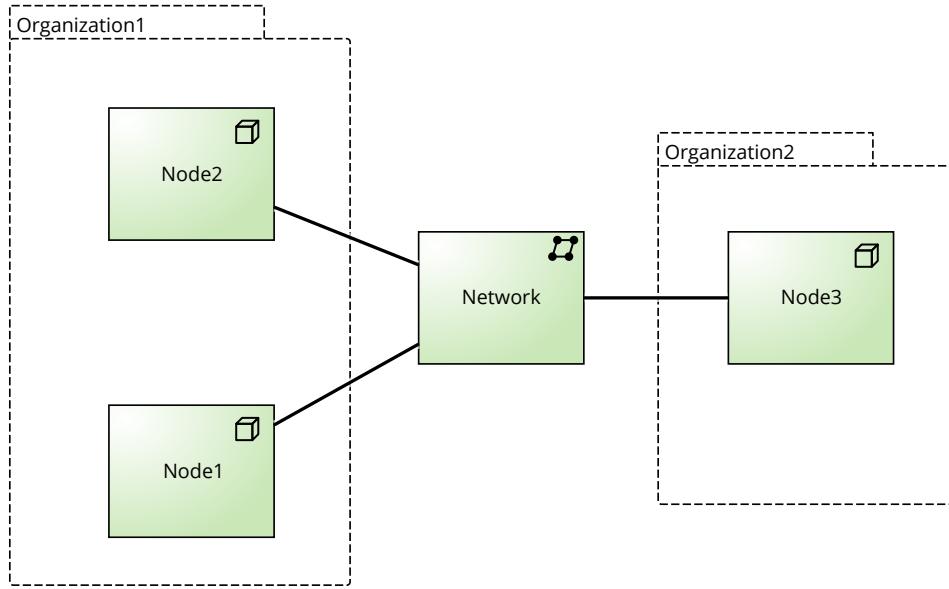
- Aggregate elements together
- Aggregate elements of the same (external) organization



- We group external organizations, while we usually not group the target organization for clarity



Grouping – example: two different organizations

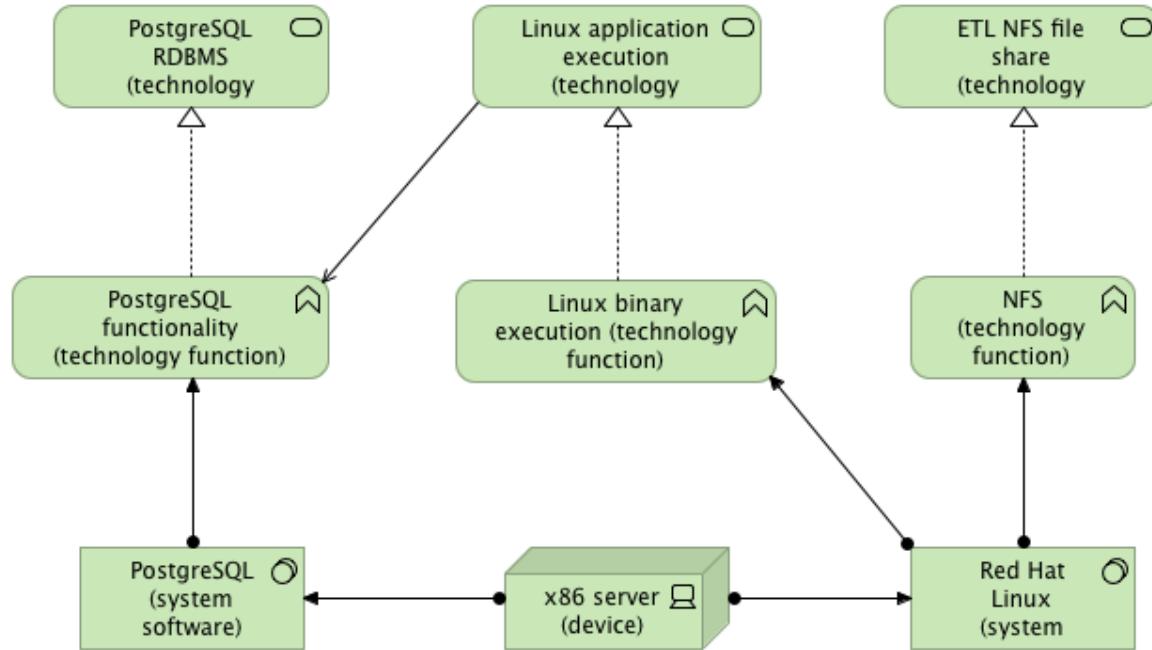




Using a node to encapsulate infrastructure

- It is possible to use a node to model an hardware and software infrastructure more or less detailed
- A nesting relationship can be defined through the relationships: *composition*, *aggregation*, *assignment* e *realization*
- Doing this, a node can model different abstraction levels

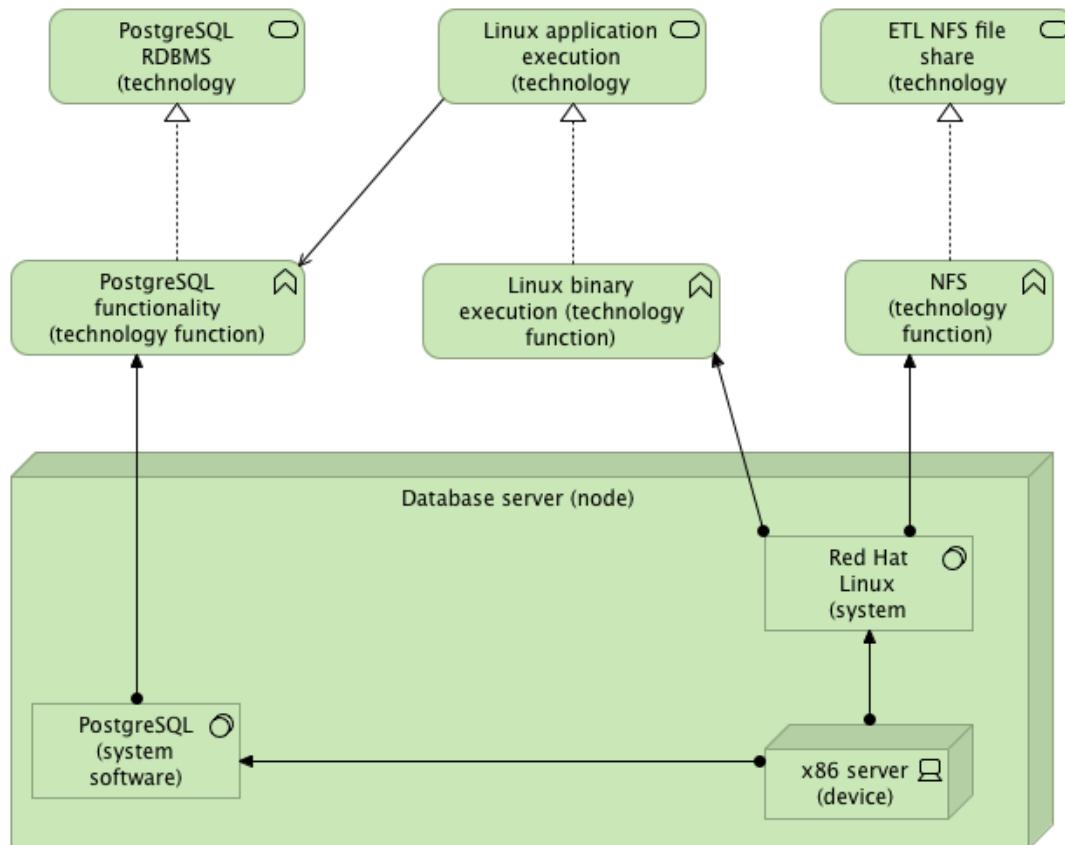
Example



- Each element is shown with all its relationships



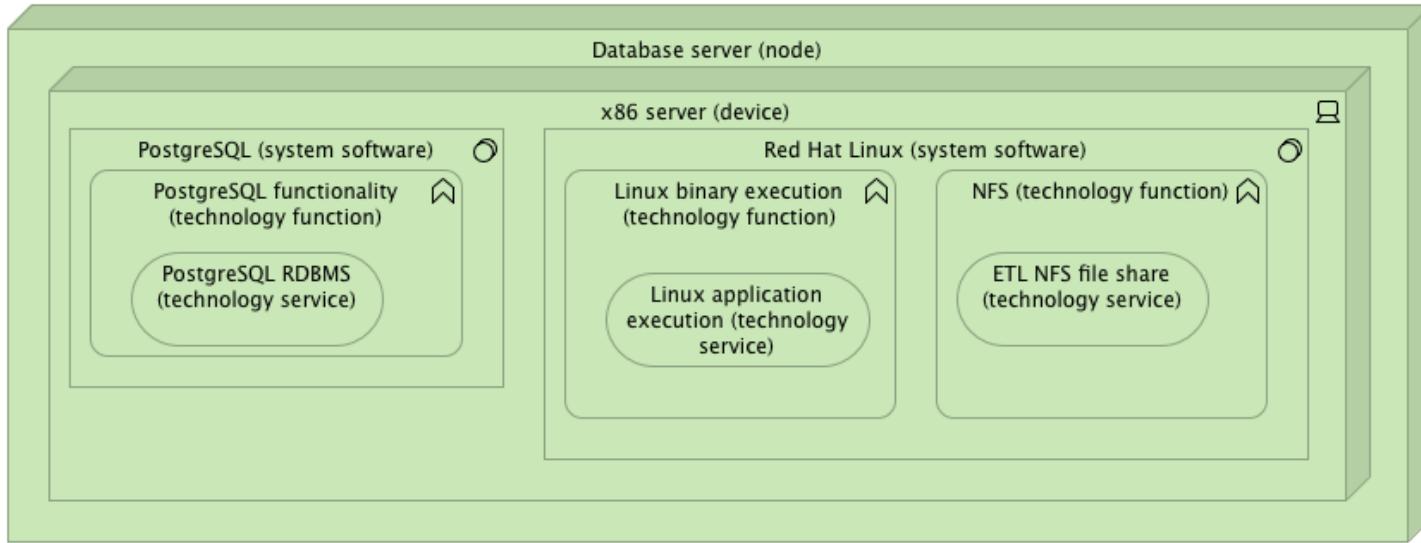
Example



- An abstract node “Database Server” is introduced and it aggregates the System Software and Device elements



Example

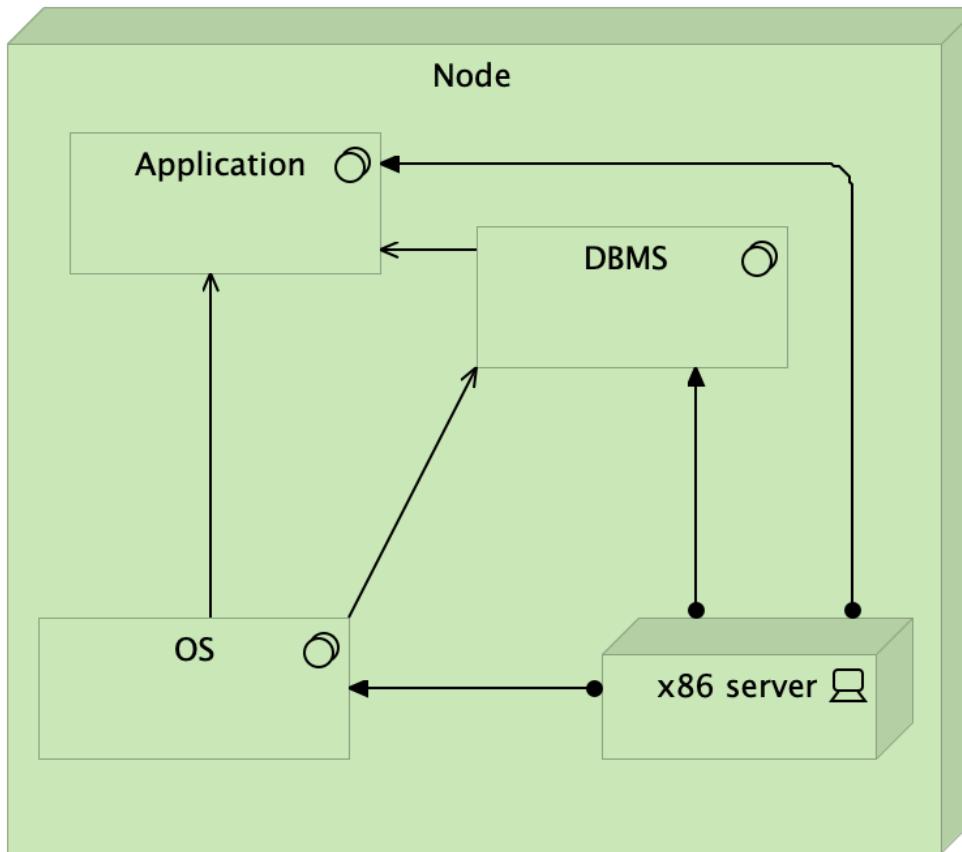


- The relationships *assignment* and *realization* allow the nesting of children elements inside parent elements
- Doing this, all the infrastructure is reduced to a single node
- In the following we will choose an intermediate abstraction level, as shown in the exercises



Node: how we model the details

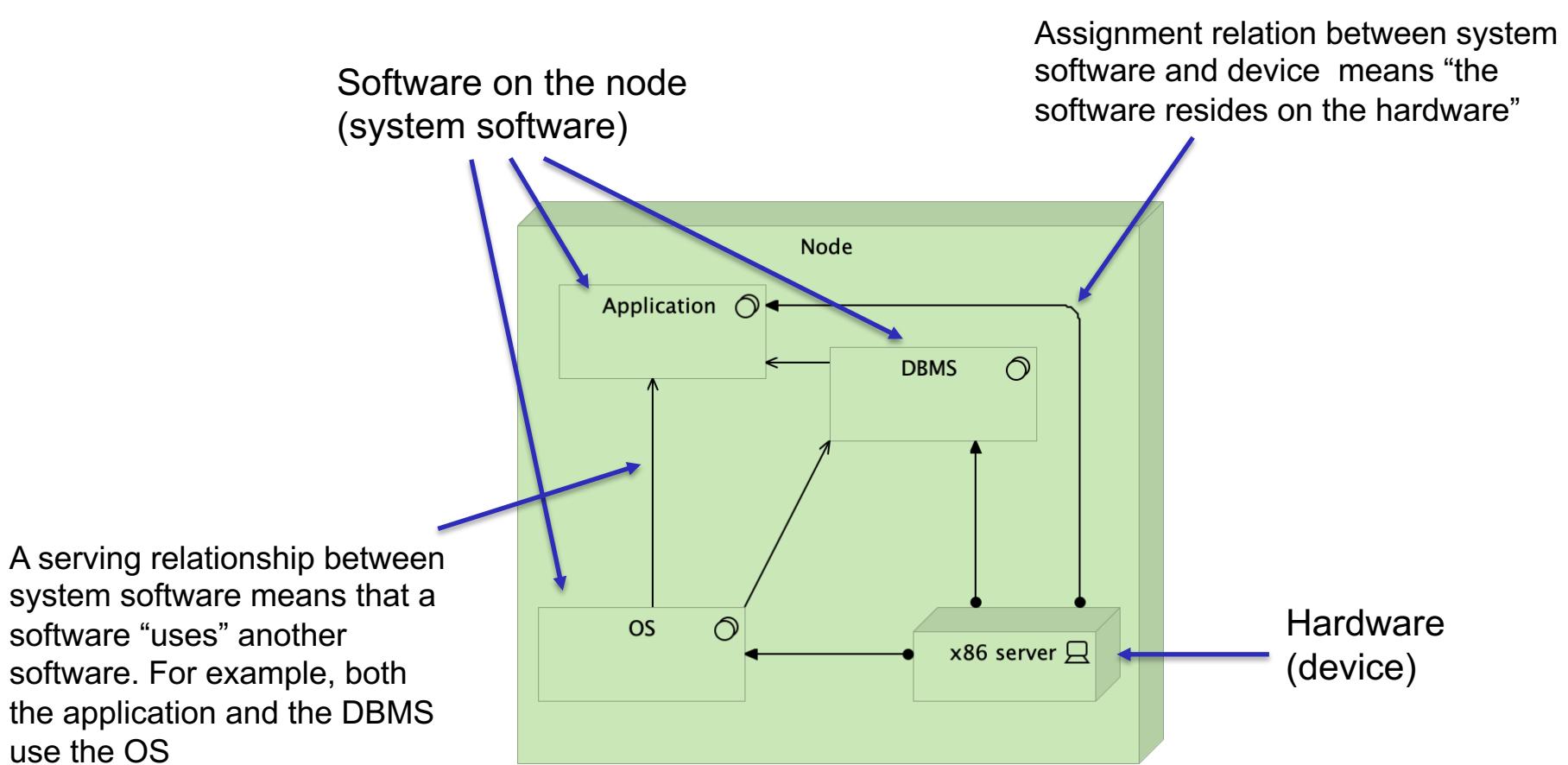
- Example of a server (node)





Node: how we model the details

- Example of a server (node)

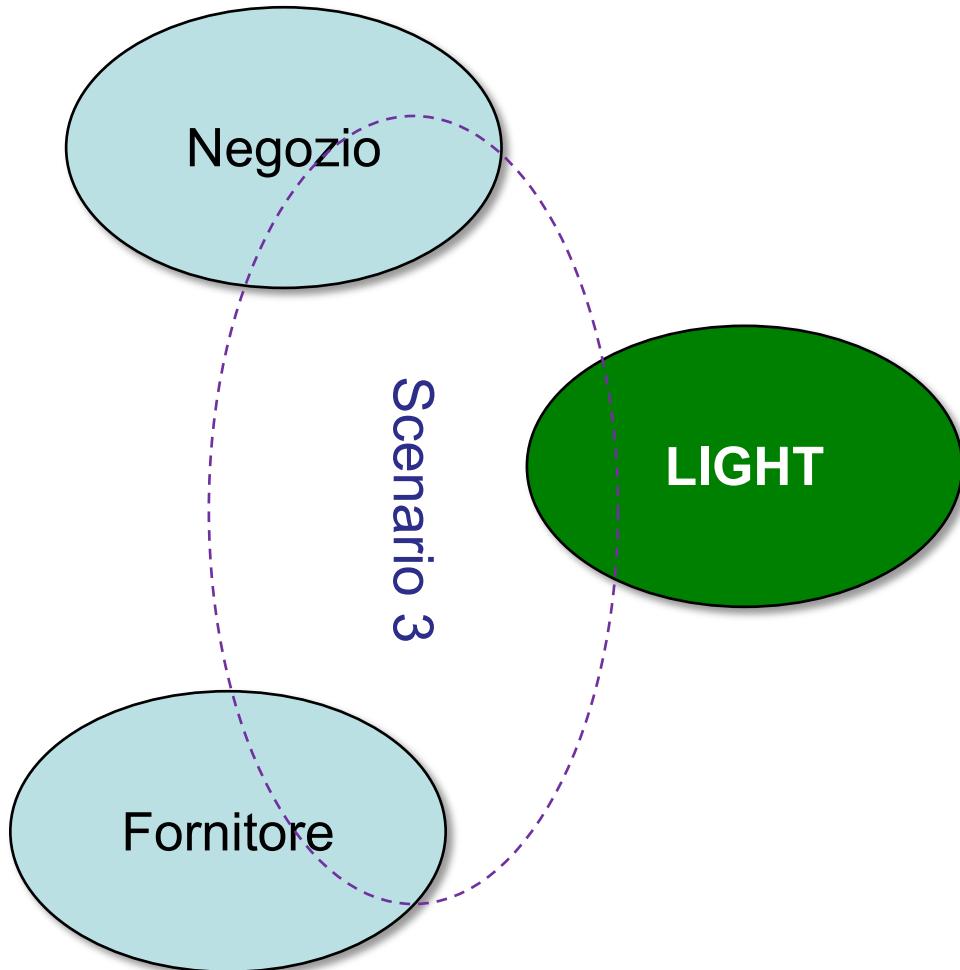




ESERCIZIO: LIGHT



Esercizio LIGHT: Business – Scenari e partecipanti





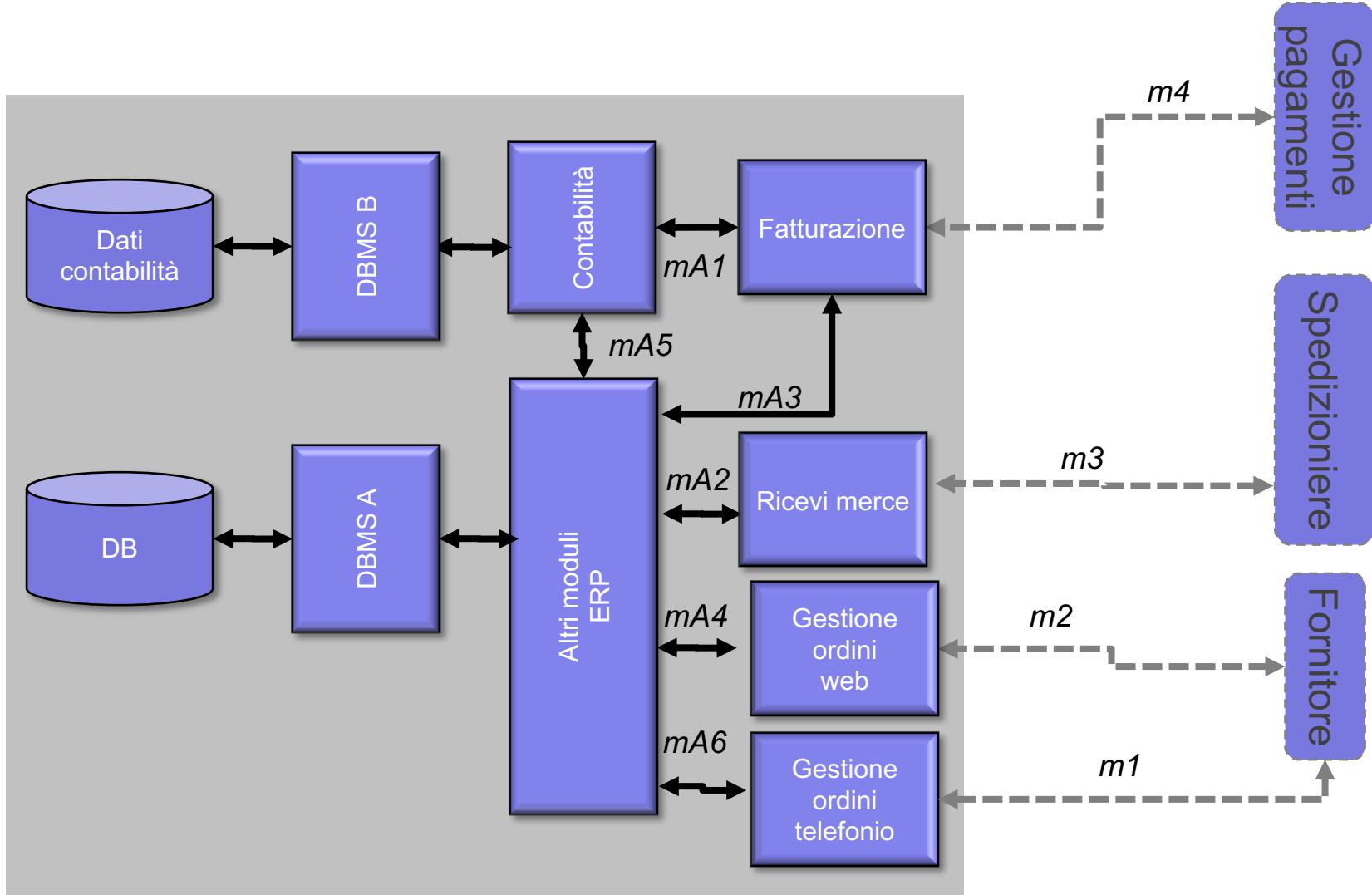
Progettazione delle soluzioni

Scenario:

LIGHT compra materie prime da fornitori e rivende a negozi

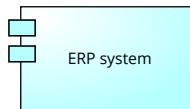
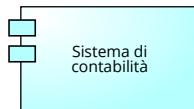


Esercizio LIGHT: Architettura party level





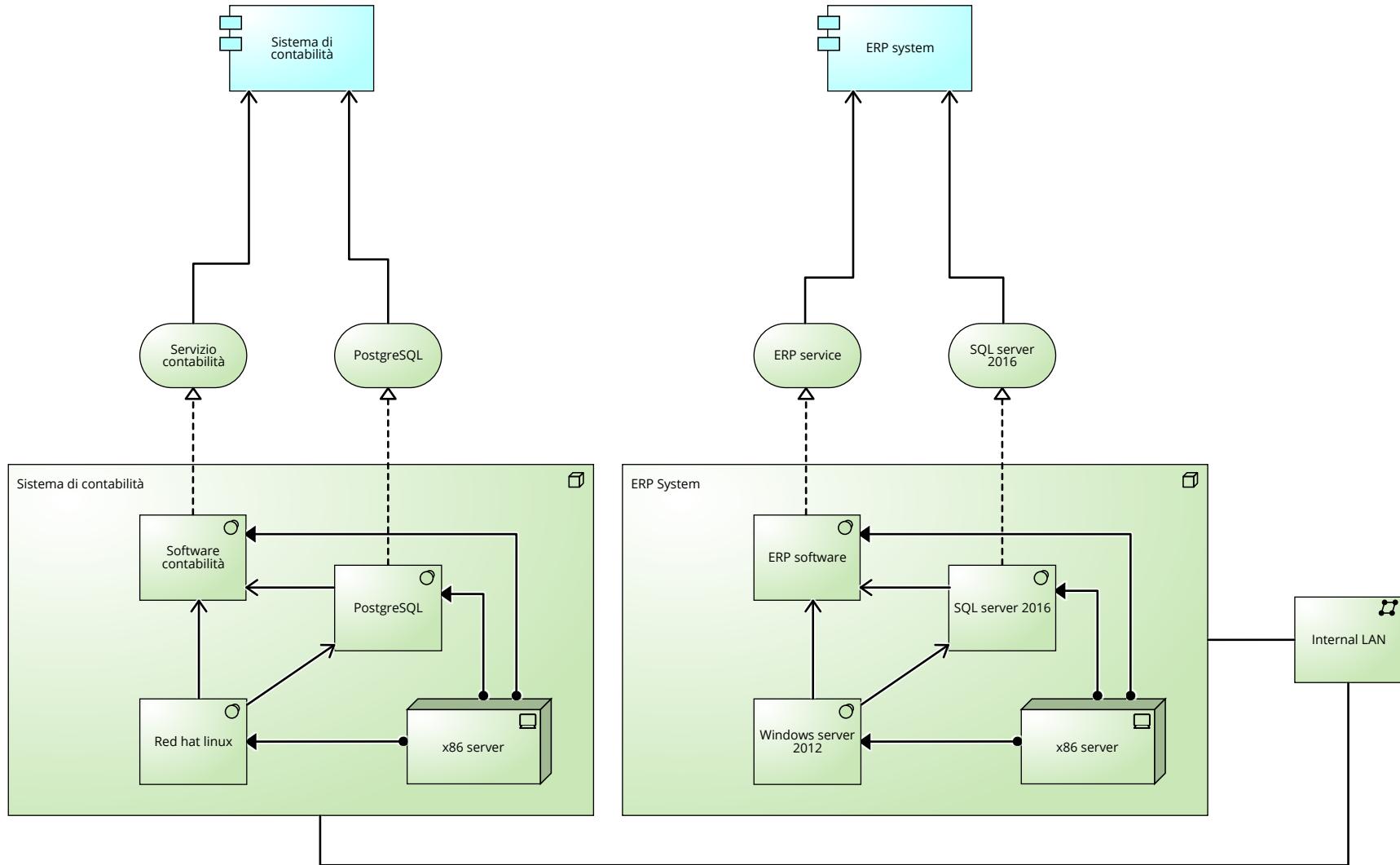
Esercizio Light: Technology level (ARCHIMATE) – STARTING POINT



- Starting from the party level diagram, each back-end component translates into an Archimate application component
- In this case, we have 2 application components
- We could have other external application components



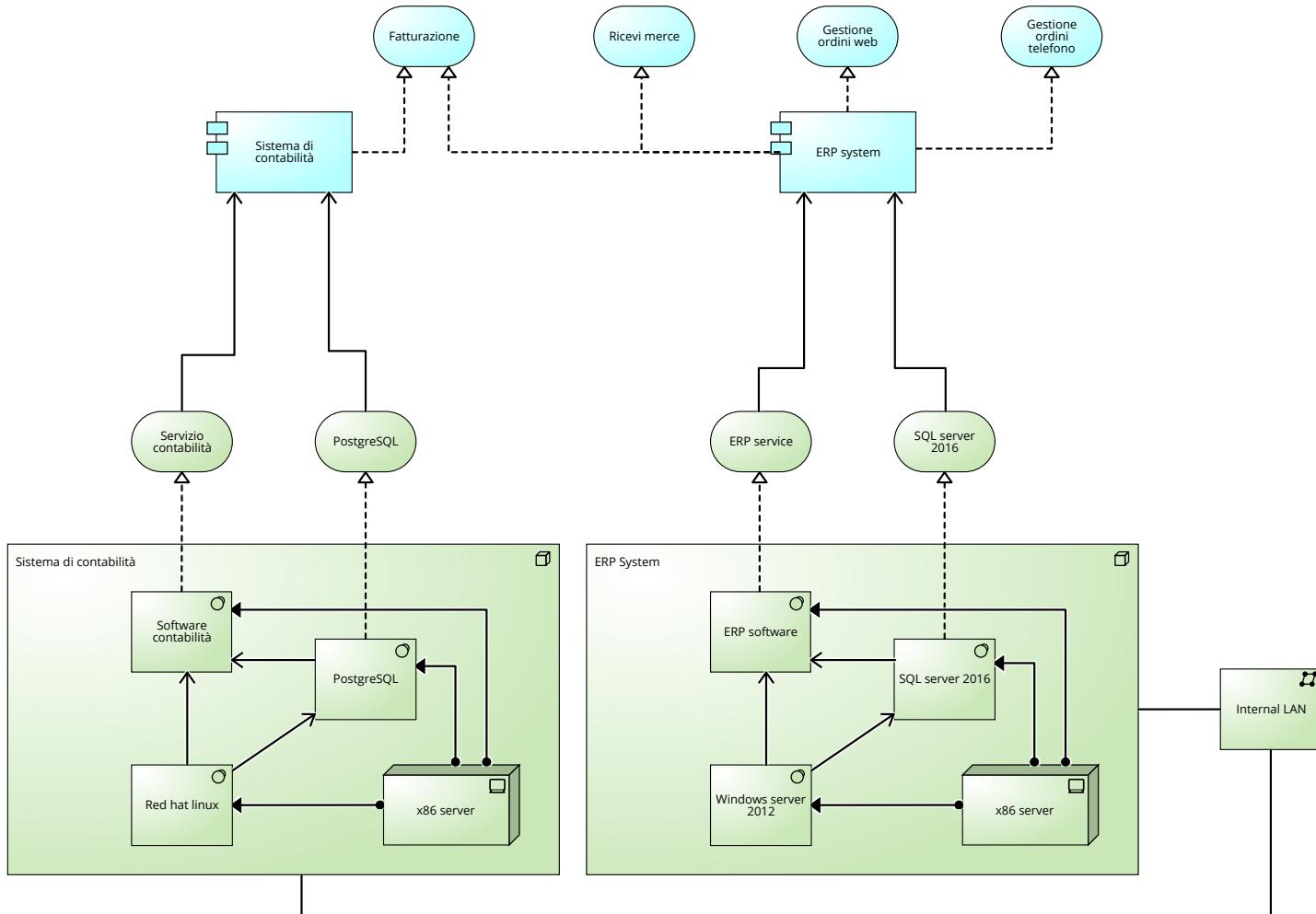
Esercizio Light: Technology level (ARCHIMATE) – on premises (part 1)





Esercizio Light: Technology level (ARCHIMATE) – on premises (part 1)

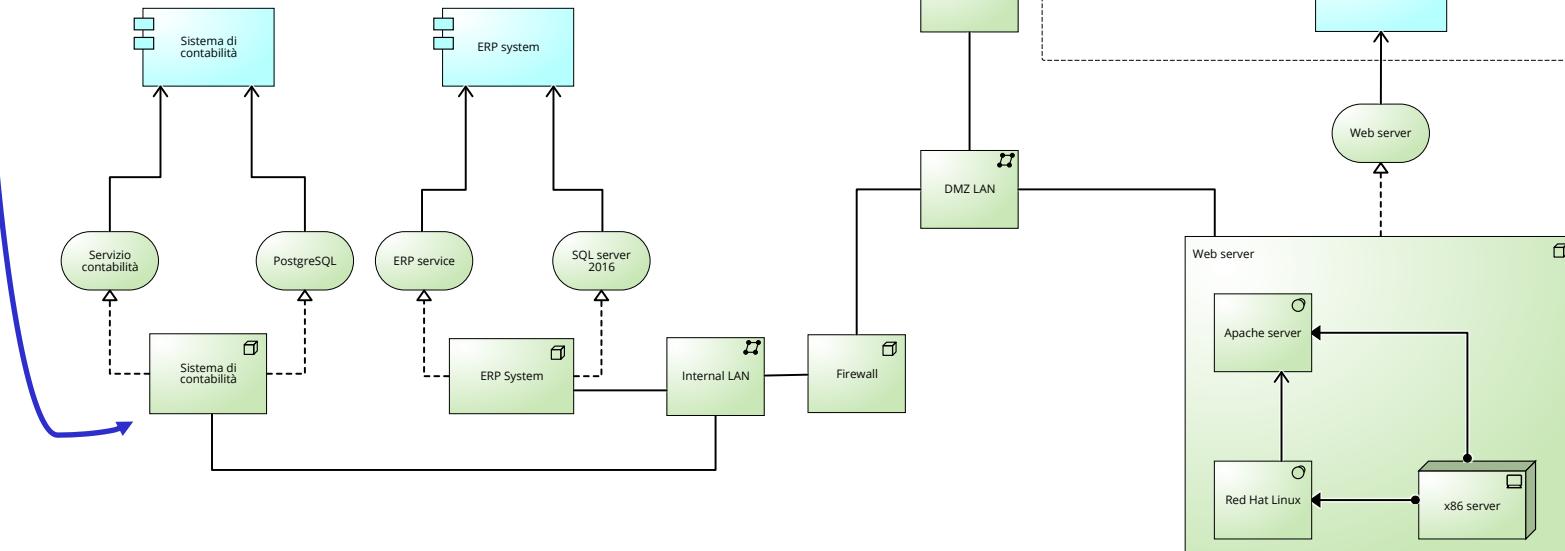
- With the front-end services seen in BOAT (not required, just for reference)
- They are represented as application services





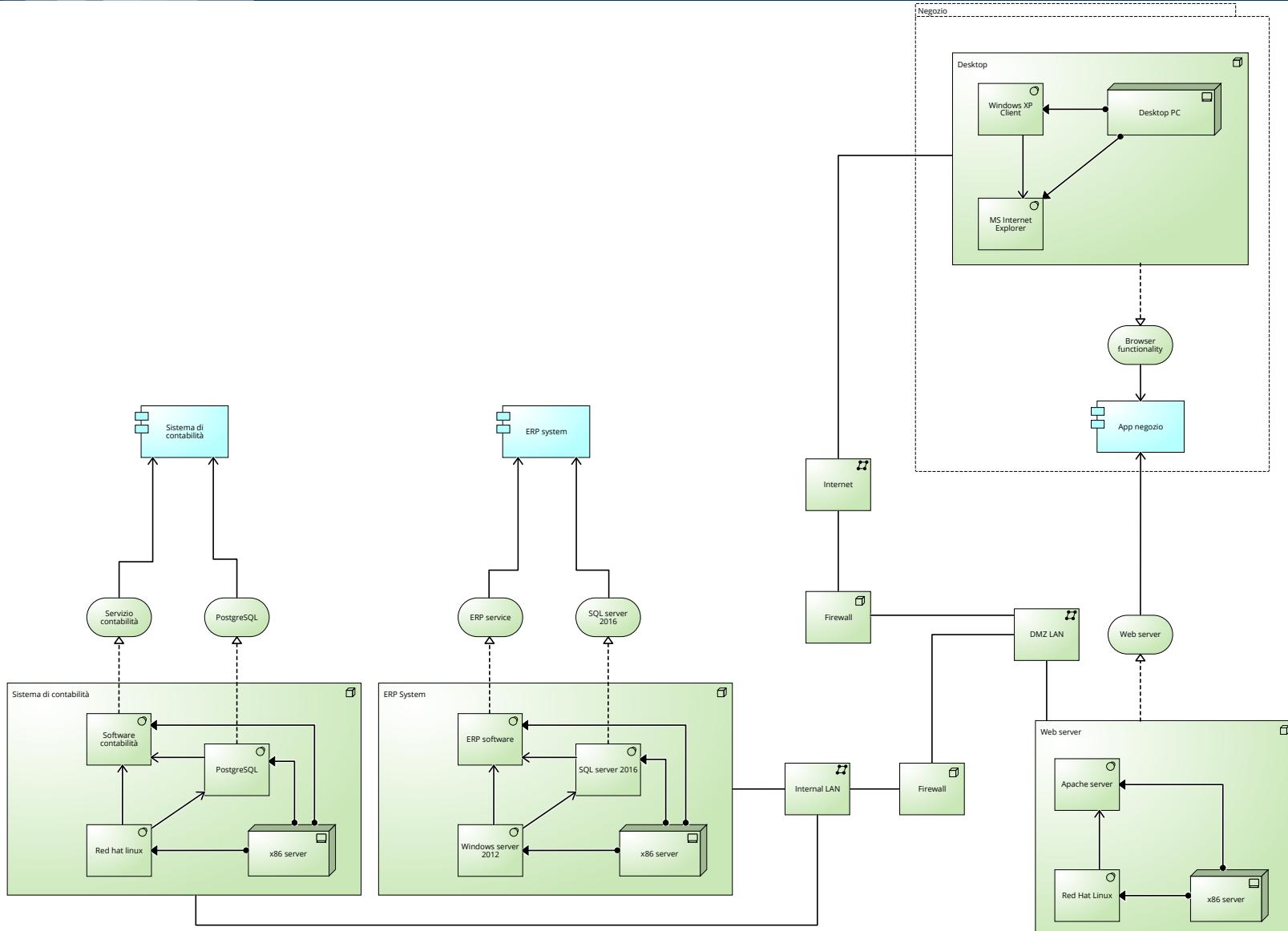
Esercizio Light: Technology level (ARCHIMATE) – on premises (part 2)

Some nodes details are hidden for compactness of the slide





Esercizio Light: Technology level (ARCHIMATE) – on premises (complete)



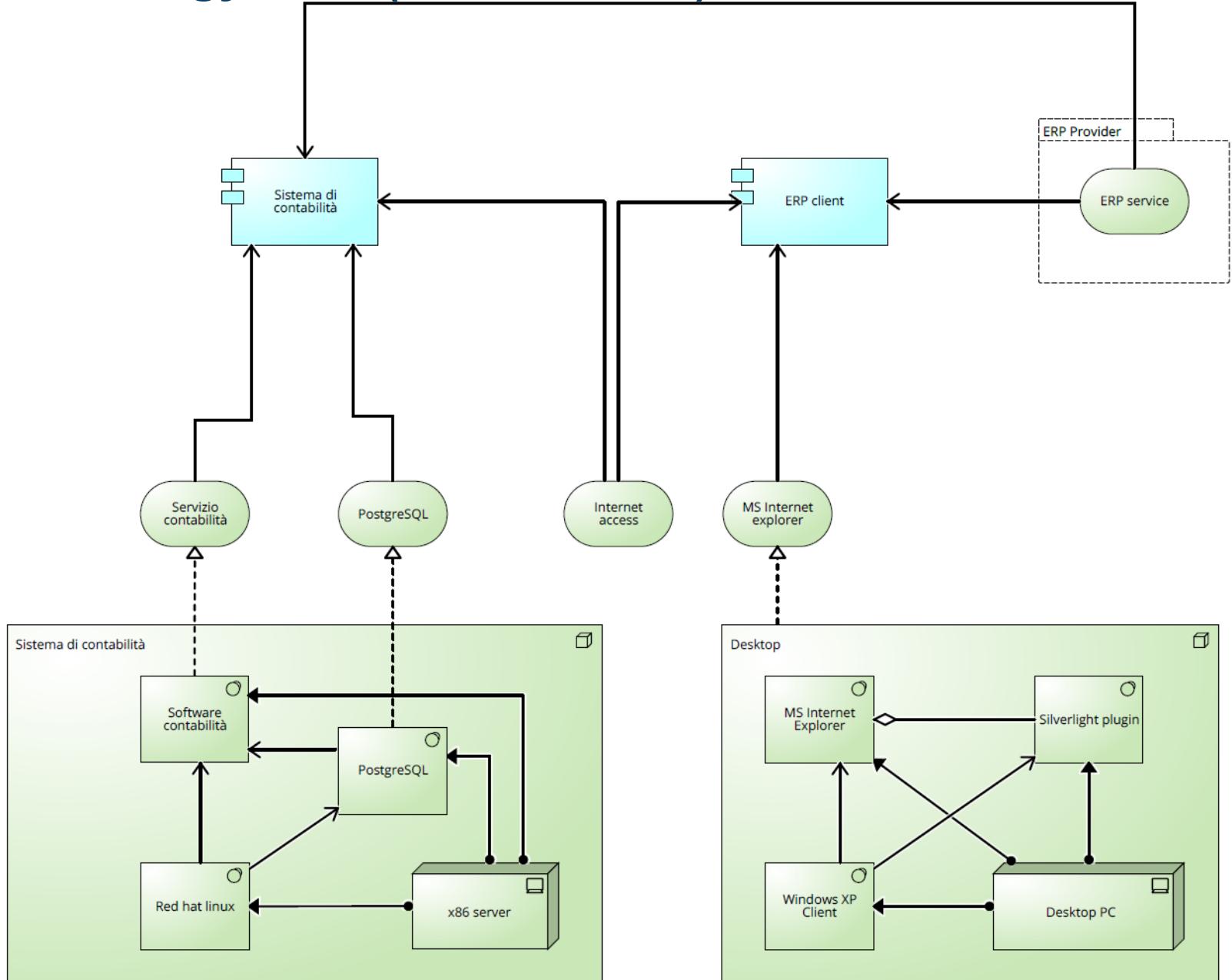


Esercizio Light: Technology level (ARCHIMATE) – on premises (complete)

- In this case we have represented both the company's infrastructure and a client
- In general, an exercise specifies what to represent



Esercizio Light: Technology level (ARCHIMATE) – SaaS, no client





Esercizio Asso (TdE 6/18)

Asso è un'azienda di assicurazioni per viaggiatori. I servizi offerti dall'azienda sono: apertura e chiusura pratiche, erogazione rimborsi, ricezione pagamento premio e segnalazione furti. Come tutte le organizzazioni Asso ha un sistema informativo che, però, supporta solo la gestione delle pratiche. Tutti i servizi di Asso vengono offerti via sportello, tranne la segnalazione dei furti che viene fatta anche via telefono. Asso vuole incrementare l'offerta di servizi al cliente e decide di offrire tutti i suoi servizi, tranne la segnalazione furti, anche online. Vuole inoltre arricchire il sistema informativo in modo da supportare tutti i servizi offerti dall'azienda. Per i pagamenti di rimborsi e premi Asso si appoggerà ad una azienda esterna. Dopo un'attenta valutazione i manager hanno deciso di optare per una soluzione make e on premise, per ridurre il rischio di esposizioni di dati sensibili degli utenti. La nuova versione del sistema informativo consisterà di due moduli legacy che gestiscono le pratiche e l'anagrafica, e due nuovi moduli si occuperanno di gestire rimborsi e premi. Definire il diagramma BOAT party level di Asso.

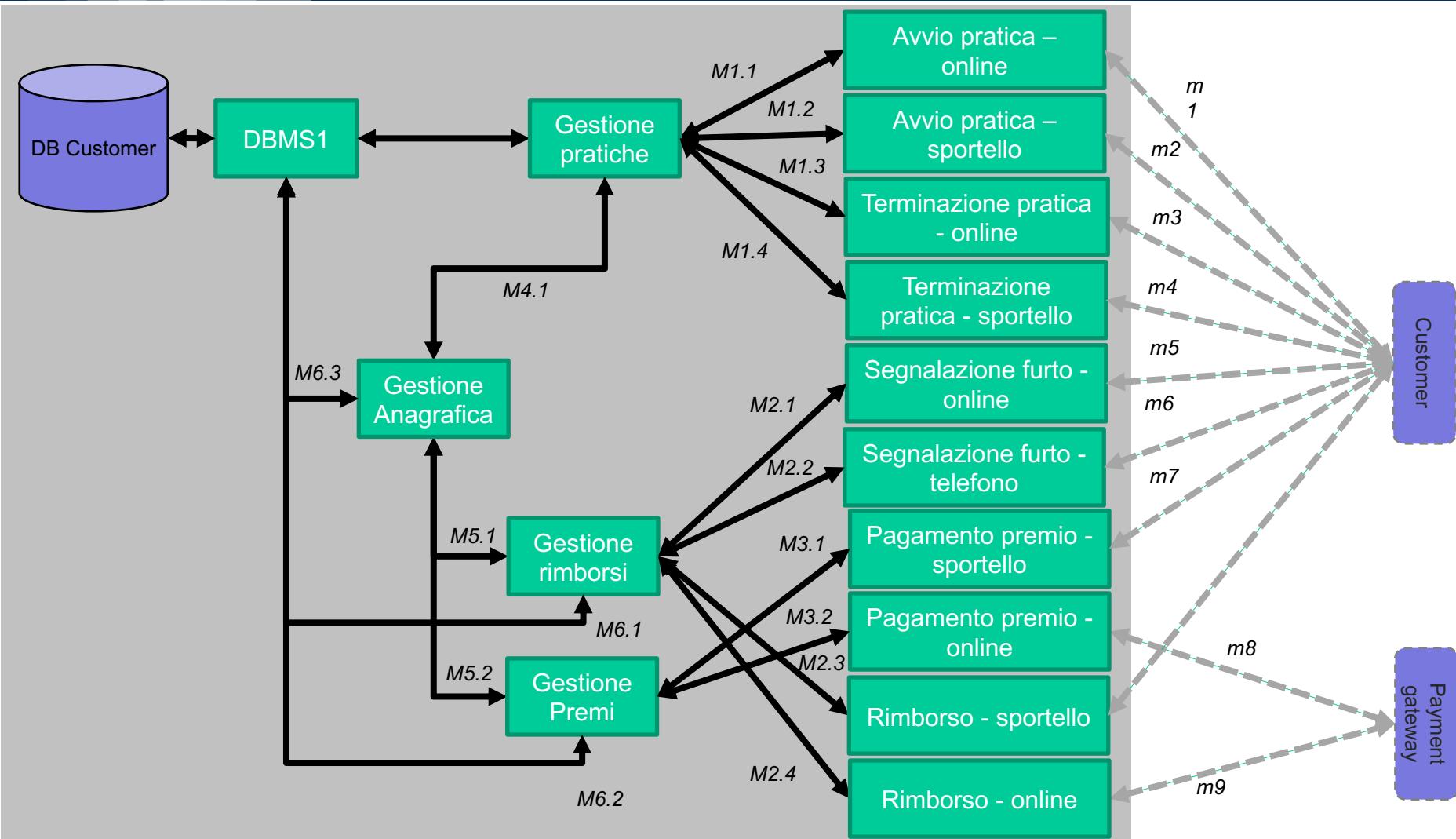
A livello architettonico, i nuovi applicativi risiedono sullo stesso server fisico, che si affianca al sistema legacy, che a sua volta risiede su un unico server insieme al suo database. I due server sono connessi tramite una LAN. I servizi forniti mediante interfaccia web si avvalgono di un web server (non è necessario rappresentare i web client). Il personale dell'azienda è dotato di un applicativo desktop connesso alla rete ad-hoc protetta da un firewall.

Con riferimento al testo sopra riportato, progettare, utilizzando l'approccio BOAT:

- il modello party level
- il modello Archimate, indicando i componenti applicativi e l'infrastruttura tecnologica.

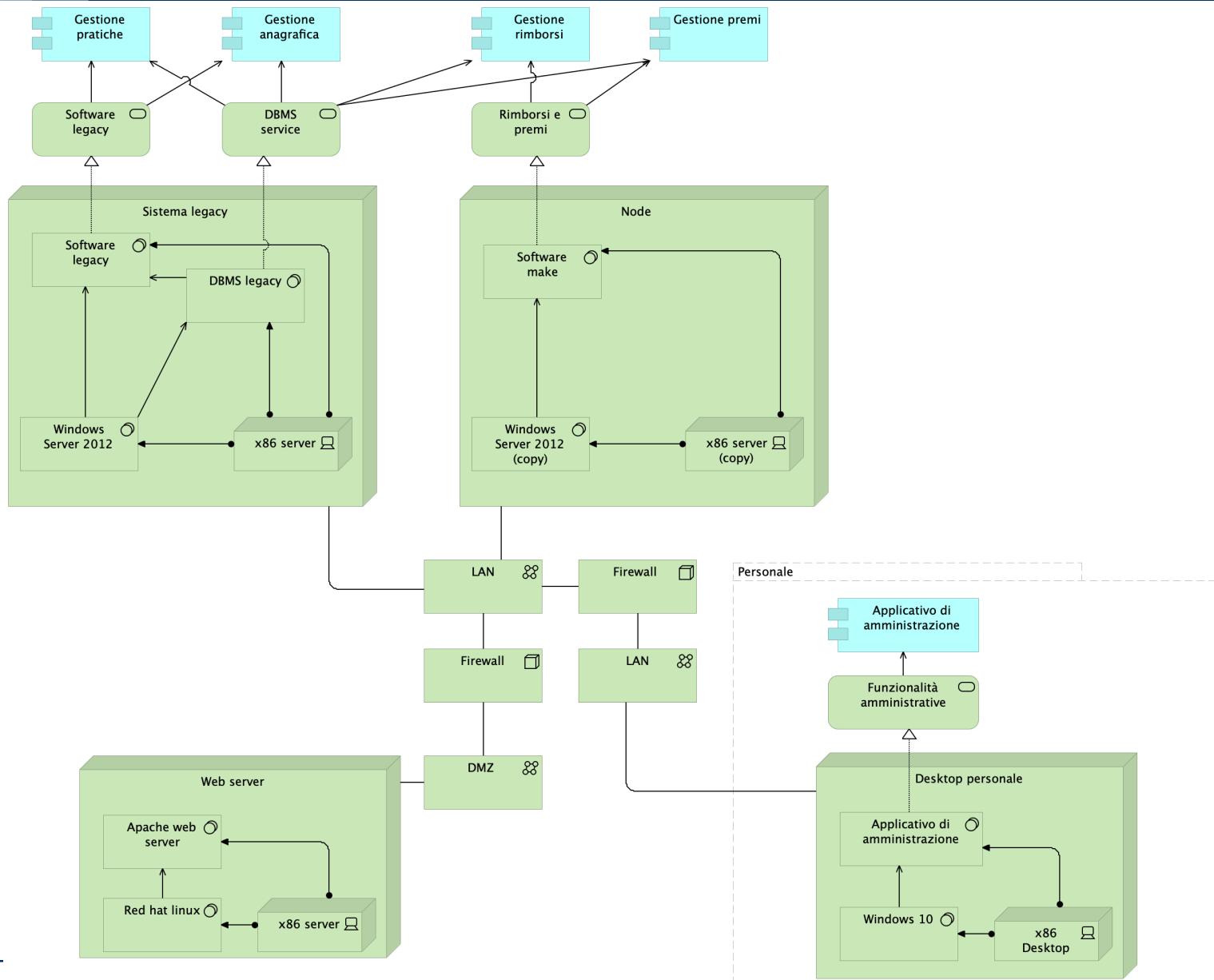


Esercizio Asso (TdE 6/18) – Party level





Esercizio Asso (TdE 6/18) – Archimate





References

- Archimate 3.0.1 specification
- Wierda: *Mastering ArchiMate*
- Lankhorst et al.: *Enterprise Architecture at Work*