Avancier

Avancier Methods (AM)
Software Architecture Diagrams

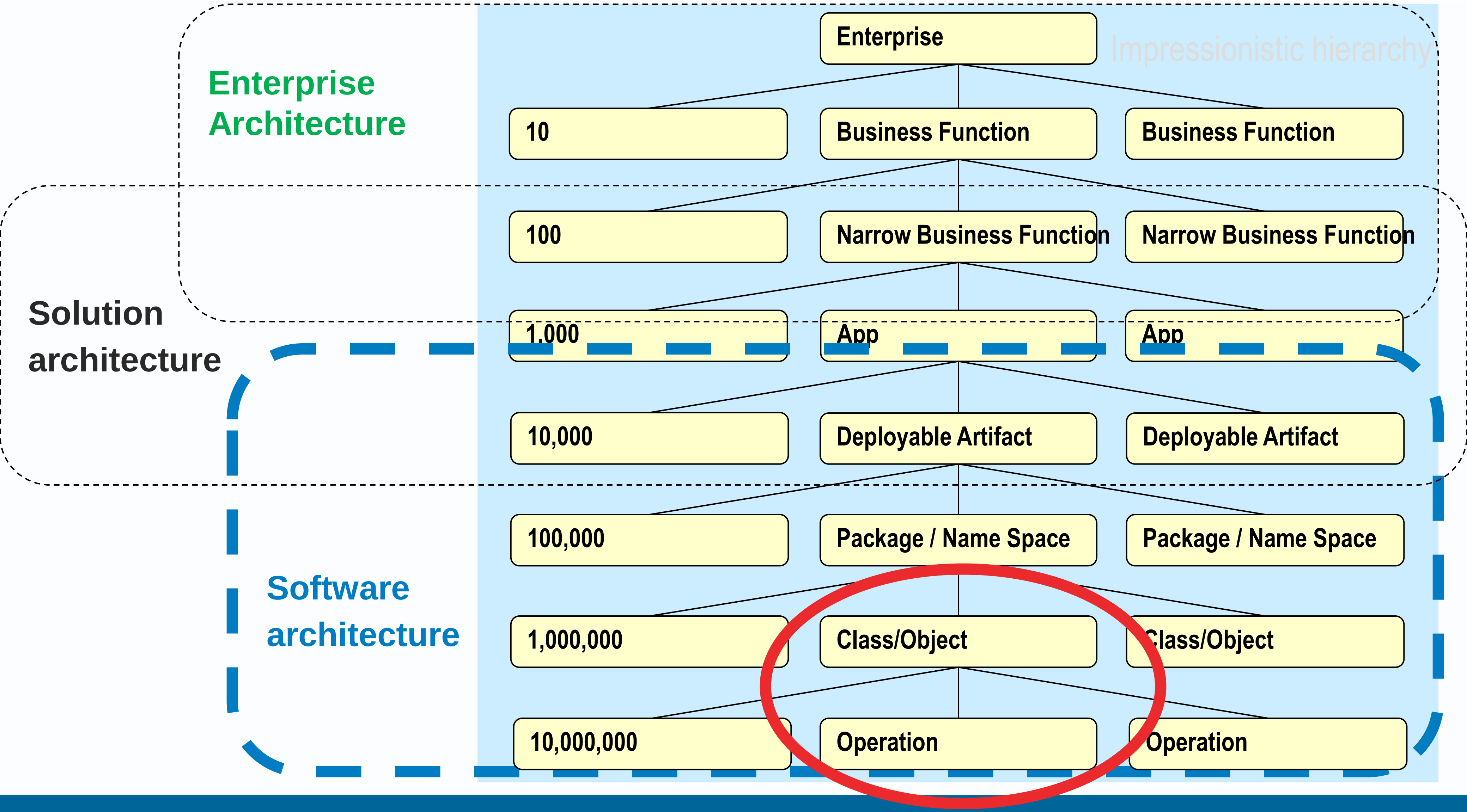
in the AM viewpoint library

ARE YOU READING THIS PAPER OUT OF CONTEXT?

Avancier

- For the solution and application architecture context, go to http://avancier.website

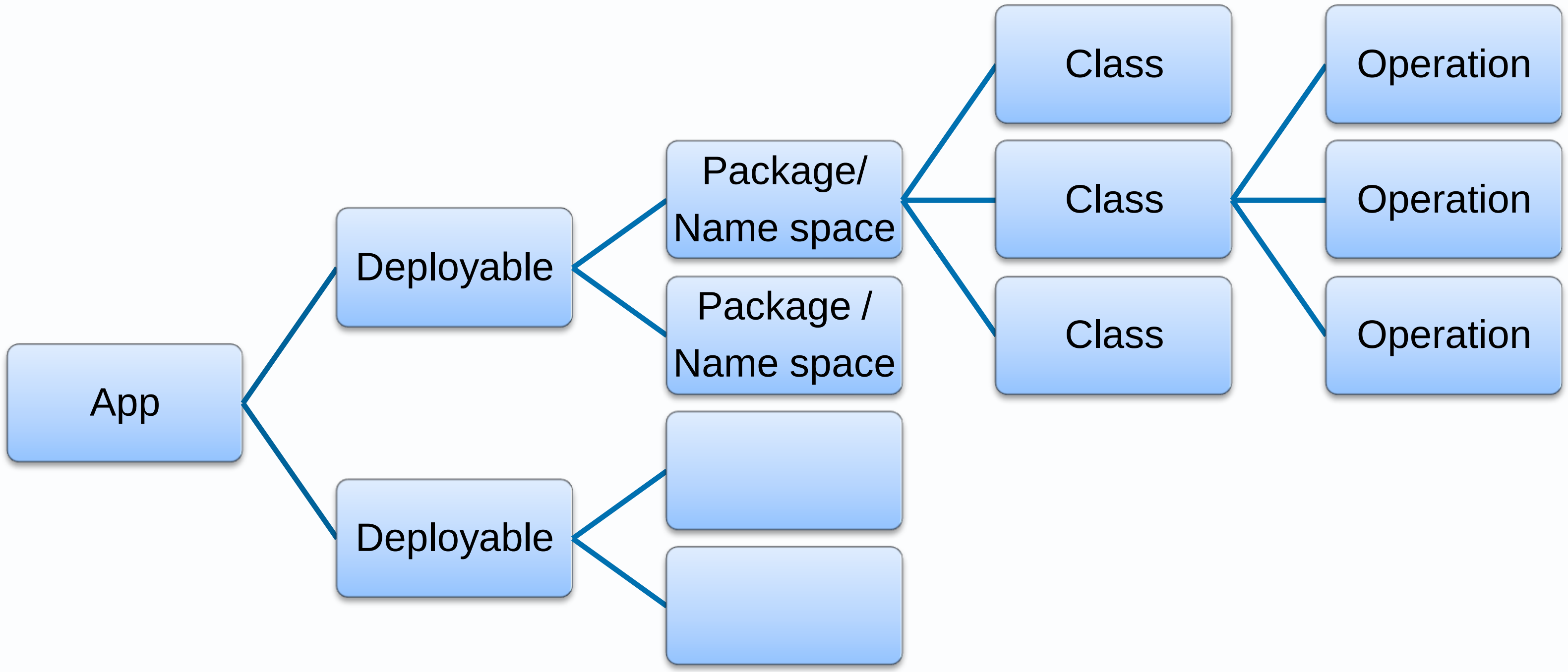
We’re at a level of granularity below EA



Software architecture catalogues

- Service catalogue (w Qualities of a Service)
- Component catalogue

- You decide the granularity



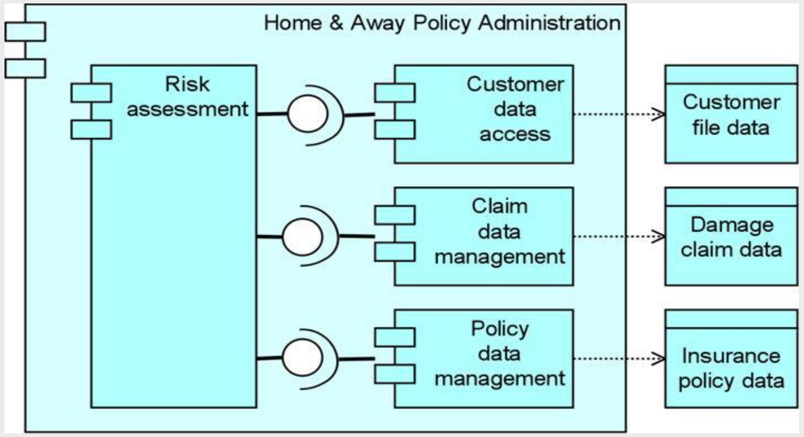
Software architecture diagrams

- Software Engineering diagram (TOGAF)
- Software Layering diagram
- Component Dependency diagram
- UML diagrams include
  - UML activity diagrams
  - UML use case diagrams
  - UML class diagrams
  - UML sequence diagrams
  - UML state machine diagrams

Software Engineering Diagram

Avancier

- breaks applications into packages, modules, services, and operations from a development perspective.
- enables more detailed impact analysis when planning migration stages, and analyzing opportunities and solutions.
- ideal for application development teams and application management teams when managing complex development environments.

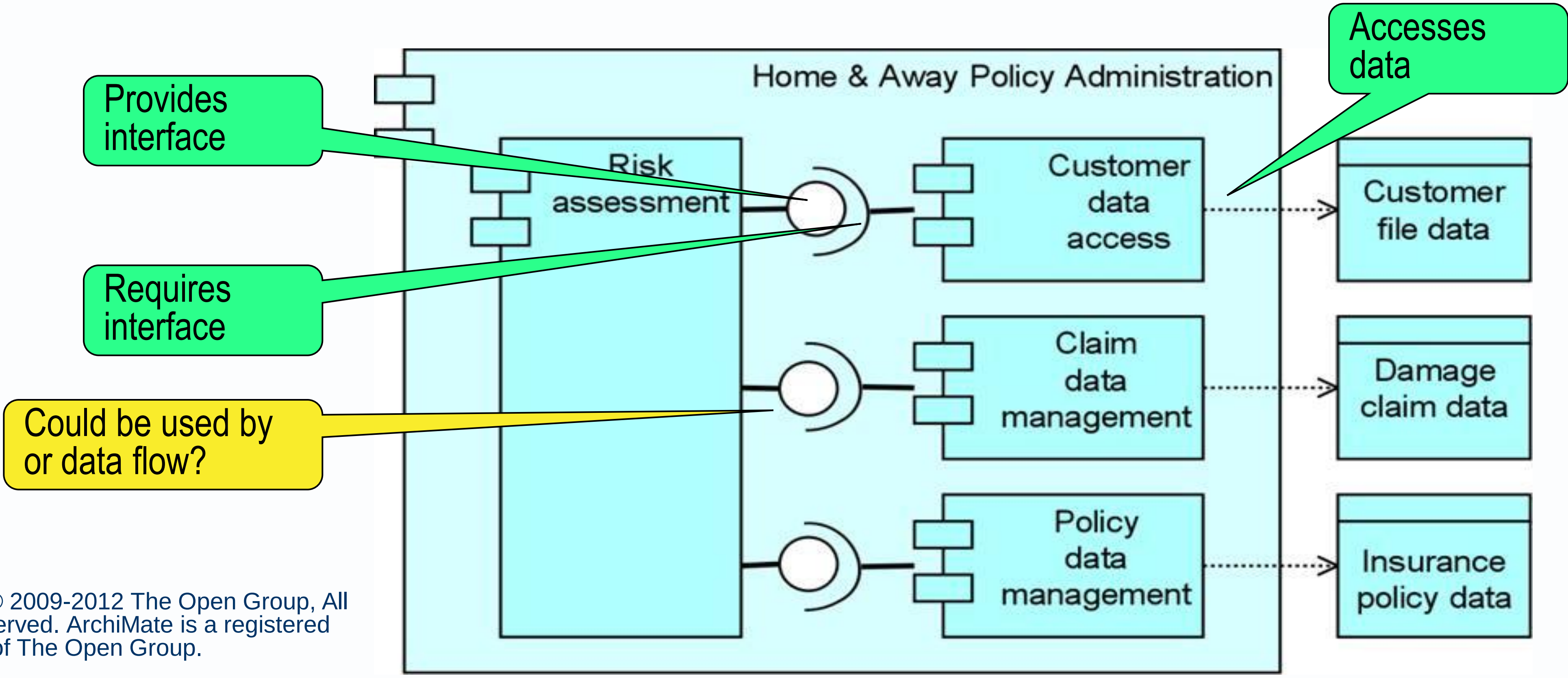


ArchiMate: Application Structure Viewpoint

Cf. SW engineering OR Application communication diagrams

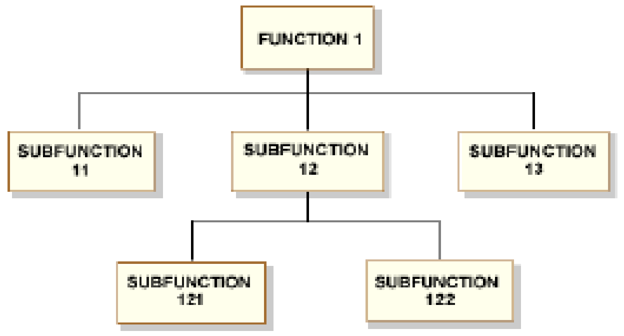
Avancier

- Stakeholders: Enterprise, process, application, and domain architects
- Concerns: Application structure, consistency and completeness, reduction of complexity

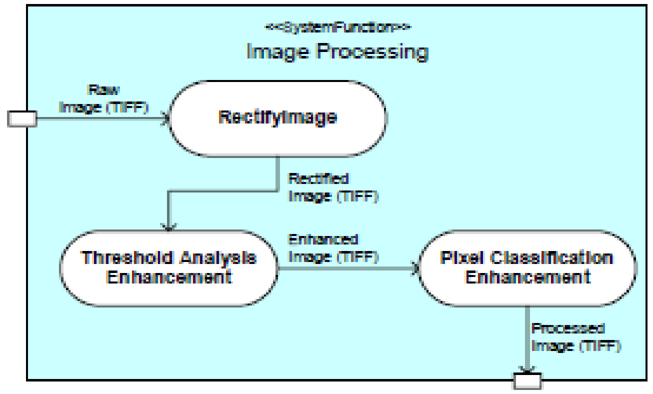


Software Engineering diagram: MODAF style

SV-4 Systems Functionality Description



Example – Generic System Functional Breakdown



Example – Hypothetical UML Activity Diagram

	Data objects: System function / sub function (including external system functions) System data flow System function hierarchy **Usage:** System analysis Specification of system functional requirements in SRD
Description: Documents system functional hierarchies, system functions, and the system data flows between them	**Alternative Views:** UML Activity Diagram

Software Layering diagram

- How is an application divided into client-server layers?
- What does each layer do?
- How does it communicate with other layers?

Generic 8-Layer scheme	Layers	Technology
UI component	UI components	HTML
UI event	UI event controllers	
UI session	UI session manager	Session bean
Transaction control		
Business services	Business service controller	Session bean
Business entities	Data access objects	Session bean
Data abstraction	Data abstraction	SQL
Database	Database	Oracle

Software Layering diagram: an illustration

Software distribution layering diagram / table

Tier	Middleware	Software layers	Language/standard
Client		Graphical user interface	Java
	Middleware	Proxy for IDL operations on server.	Java Beans
App Server		IDL operations on the server	CORBA-compliant IDL
			C++
	Middleware		ODBC
Data Server			ODBC
		Data Access operations	SQL
		Data storage tables	DDL

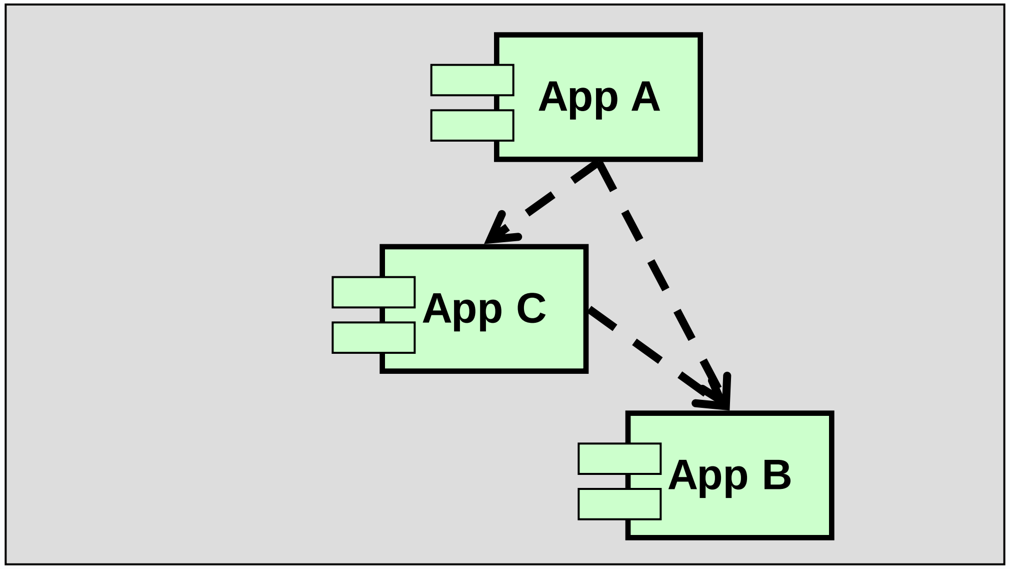
Component Dependency diagram

Avancier

- Which components require which other components?
- Useful in change impact analysis.

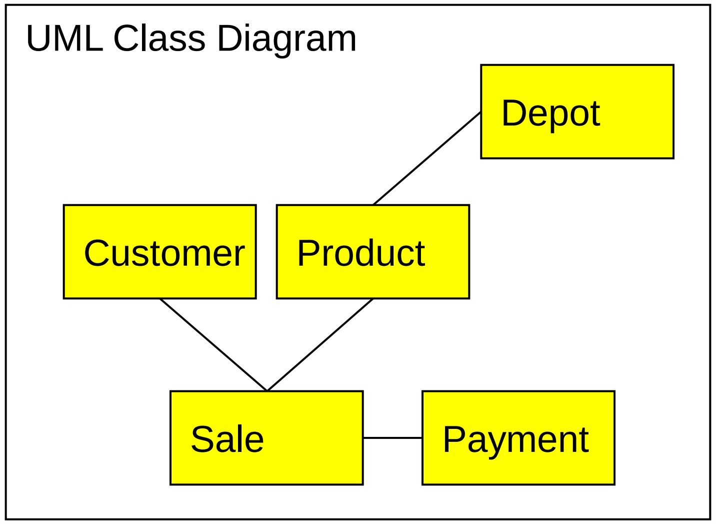
	App A	App B	App C
App A		Depends on	Depends on
App B			
App C		Depends on	

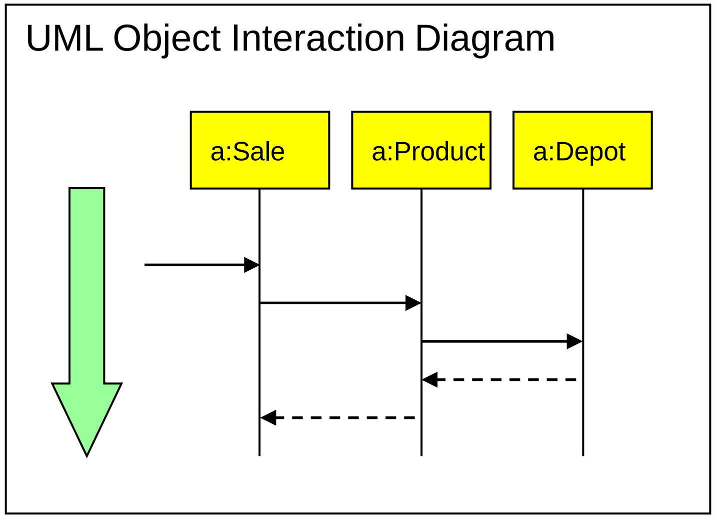
- A dependency arrow can represent several flows or service invocations.
- So, a good choice where the number of inter-component flows would be overwhelming



At http://avancier.website, see “UML distilled slide shows” for

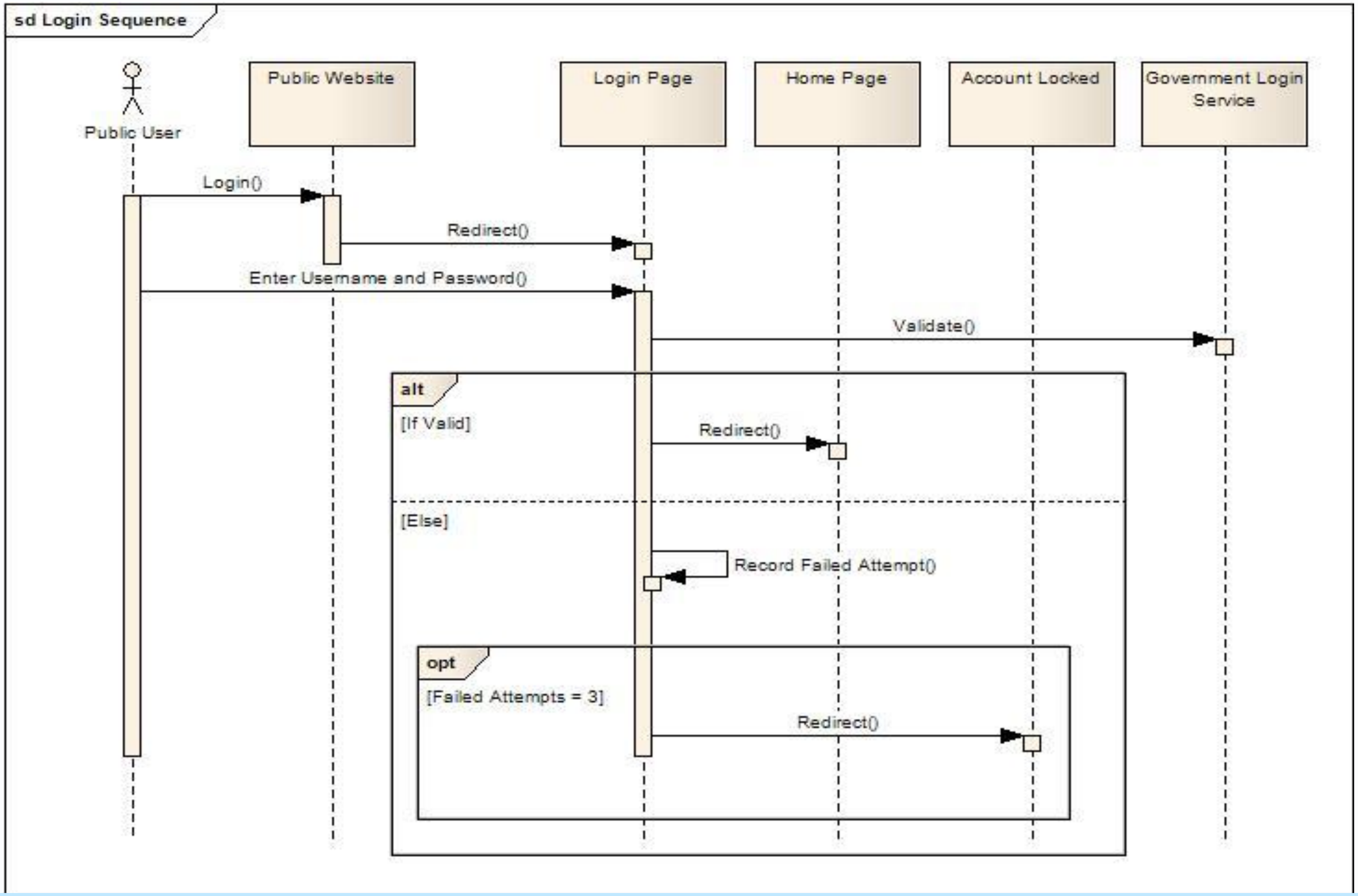
- UML activity diagrams
- UML use case diagrams
- UML class diagrams
- UML sequence diagrams
- UML state machine diagrams





UML sequence diagram: illustrating some software-level niceties

Architects’ sequence diagrams are often sketchy and feature asynchronous data flows



http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/353/Enterprise-Architect-for-Business-Analysts.aspx

At http://avancier.website, see "Technology Infrastructure" diagrams for…

Avancier

- **Application and User Location Diagram**
  - "shows the geographical distribution of applications, where applications are used by the end user; where the host application is executed and/or delivered in thin client scenarios;
  - where applications are developed, tested, and released; etc."

- **Application/Technology Matrix**
  - "documents the mapping of business systems [i.e applications] to technology platform."

- **Processing Diagram**
  - "focuses on deployable units of code/configuration and
  - how these are deployed onto the technology platform."

- **Software Distribution Diagram**
  - "shows how application software is structured and distributed across the estate…
  - shows how physical applications are distributed across physical technology and the location of that technology…
  - enables a clear view of how the software is hosted"

- **Environments and Locations Diagram**
  - "depicts which locations host which applications…
  - what technologies and/or applications are at which locations"

- **Networked Computing/Hardware Diagram**
  - "to document the mapping between logical applications and the technology components (e.g., server) that supports the application both in the development and production environments…
  - "to show the "as deployed" logical view of logical application components in a distributed network computing environment…
  - "Enable understanding of which application is deployed where in the distributed network computing environment."