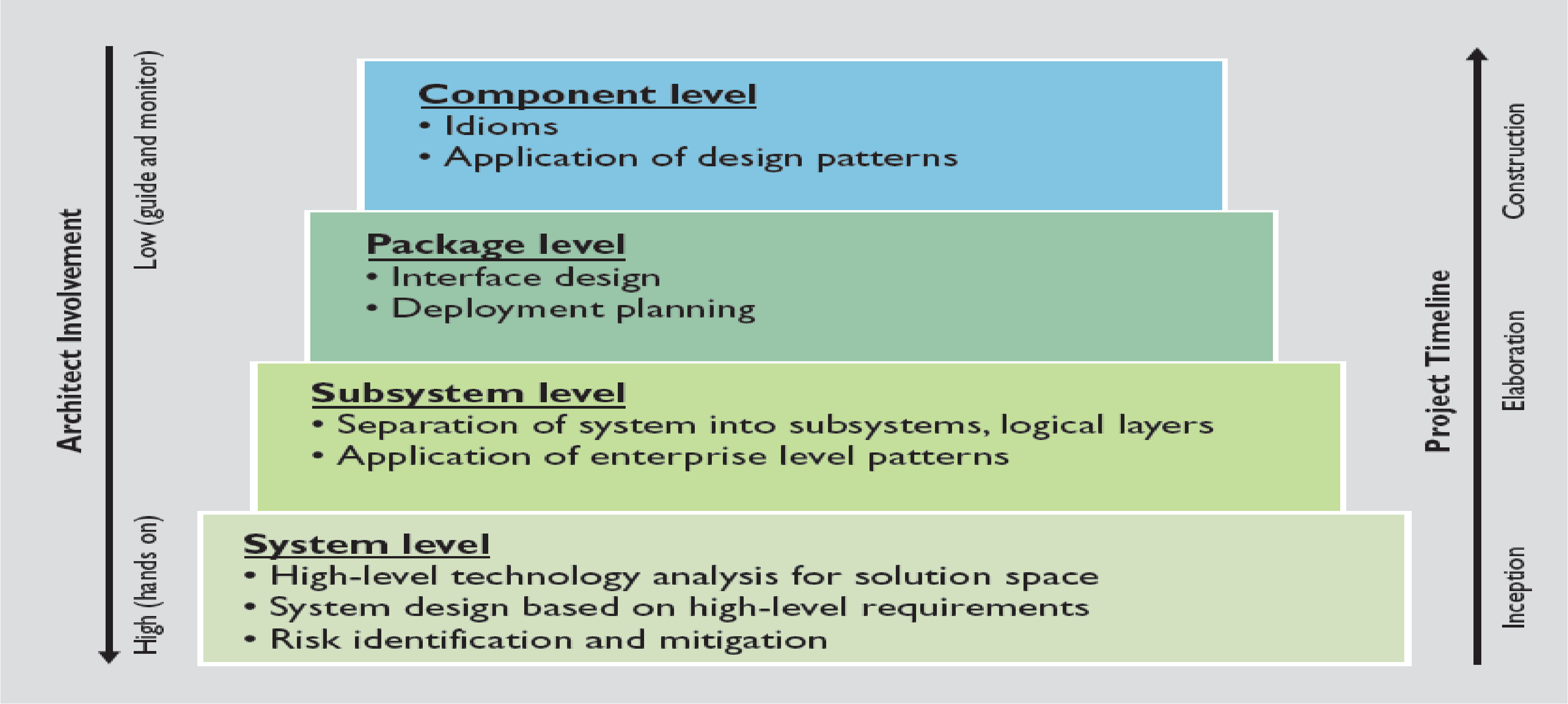
What are IT Architects and what do they do all day ?

IT Architect Roles and Responsibilities

Dr. Marcel Schlatter
IBM Distinguished Engineer
Member of the IBM Academy of Technology
marcel.schlatter@ch.ibm.com

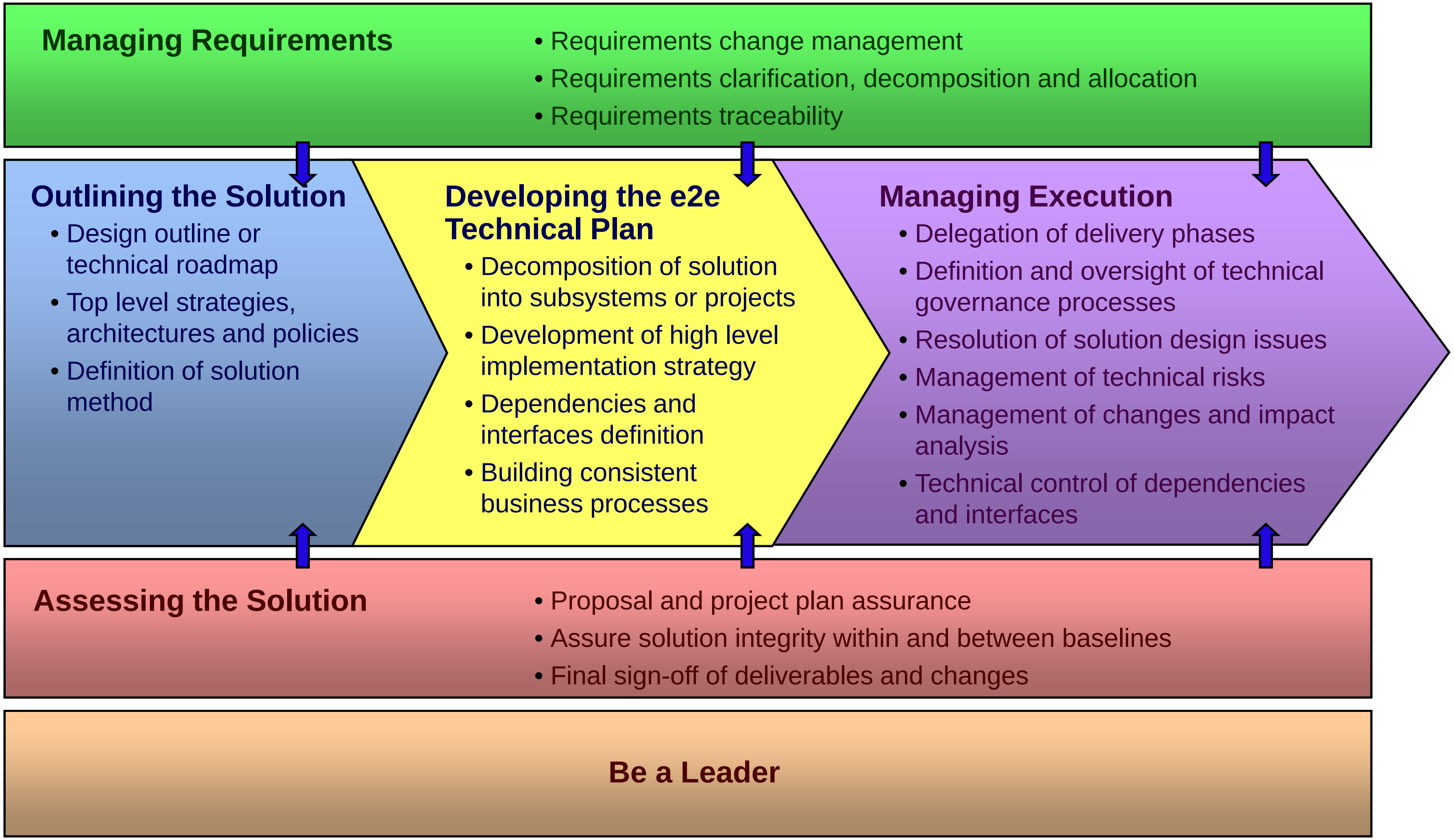
IT Architects are technically competent system-level thinkers, guiding planned and economically efficient design processes to bring a system into existence.


Architect Involvement
Low (guide and monitor)
High (hands on)
Component level
• Idioms
• Application of design patterns
Package level
• Interface design
• Deployment planning
Subsystem level
• Separation of system into subsystems, logical layers
• Application of enterprise level patterns
System level
• High-level technology analysis for solution space
• System design based on high-level requirements
• Risk identification and mitigation
Project Timeline
Construction
Elaboration
Inception

Architects focus on system- and subsystem-level issues to establish a solid foundation for detailed design, particularly for large-scale efforts

The software architect
Matthew R. McBride, Communications of the ACM
Volume 50, Number 5 (2007), Pages 75-81

Architect's Responsibilities across the full life-cycle



Managing Requirements

- Requirements change management
- Requirements clarification, decomposition and allocation
- Requirements traceability

Outlining the Solution

- Design outline or technical roadmap
- Top level strategies, architectures and policies
- Definition of solution method

Developing the e2e Technical Plan

- Decomposition of solution into subsystems or projects
- Development of high level implementation strategy
- Dependencies and interfaces definition
- Building consistent business processes

Managing Execution

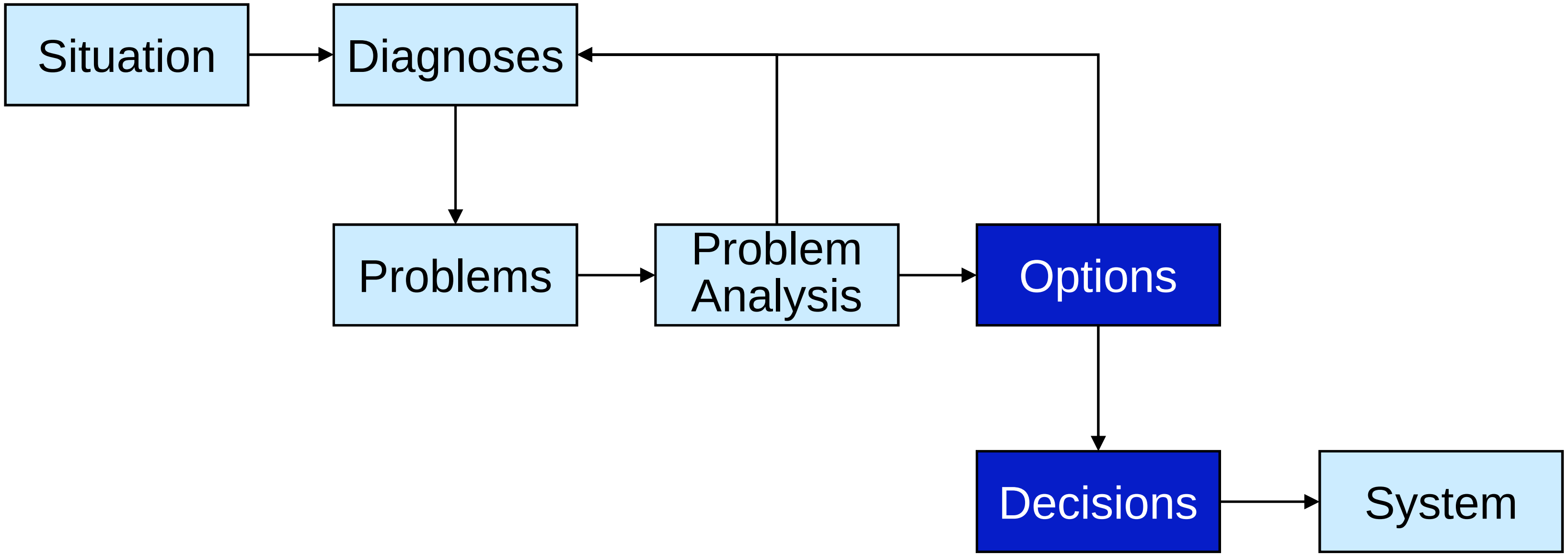
- Delegation of delivery phases
- Definition and oversight of technical governance processes
- Resolution of solution design issues
- Management of technical risks
- Management of changes and impact analysis
- Technical control of dependencies and interfaces

Assessing the Solution

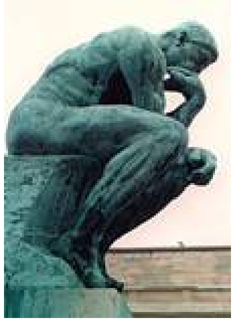
- Proposal and project plan assurance
- Assure solution integrity within and between baselines
- Final sign-off of deliverables and changes

Be a Leader

The Role of the IT Architect

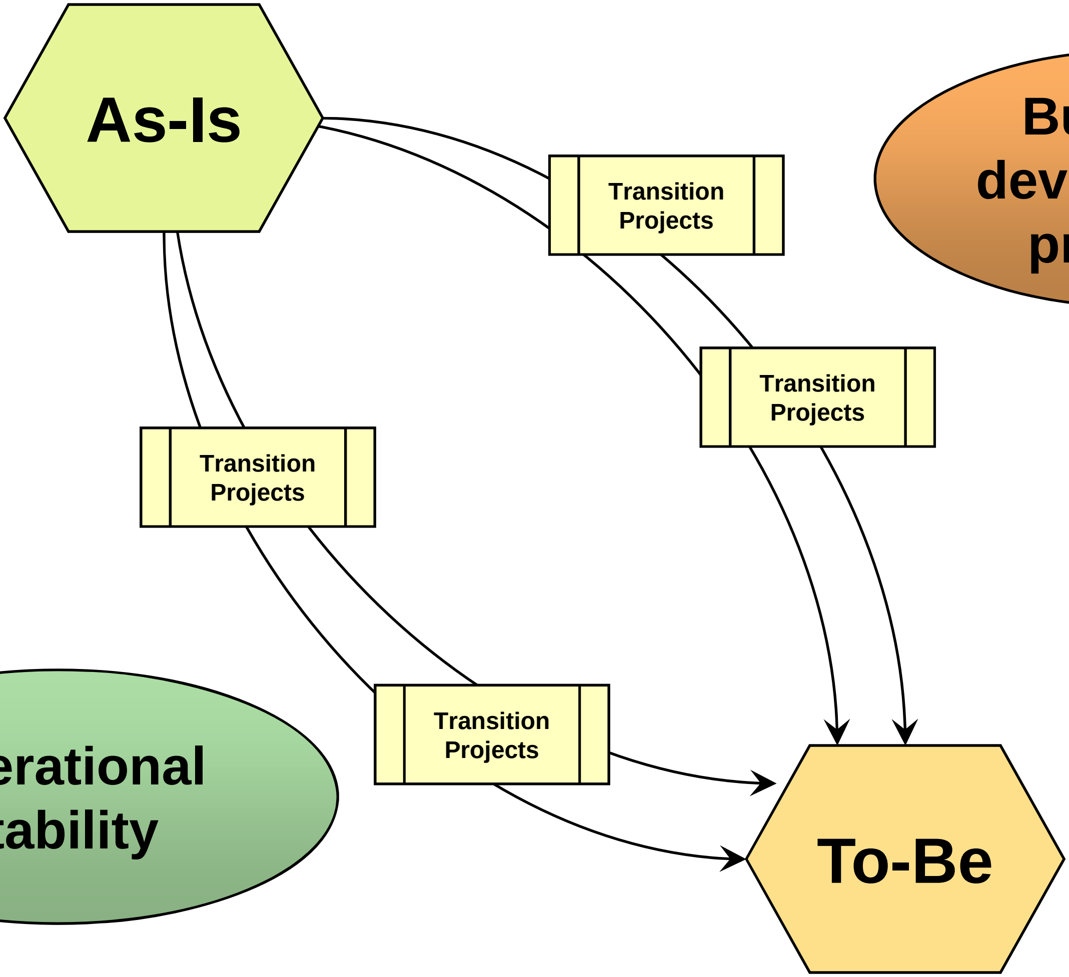


- The architect should not focus on some separate part, called the „architecture“
- The architect should assume the **responsibility** that an engineered system is optimally matched to the situation
- The architect is the author of the solution, undeniably **accountable** for the effort‘s success or failure.



Analysis

Change



Business development pressure

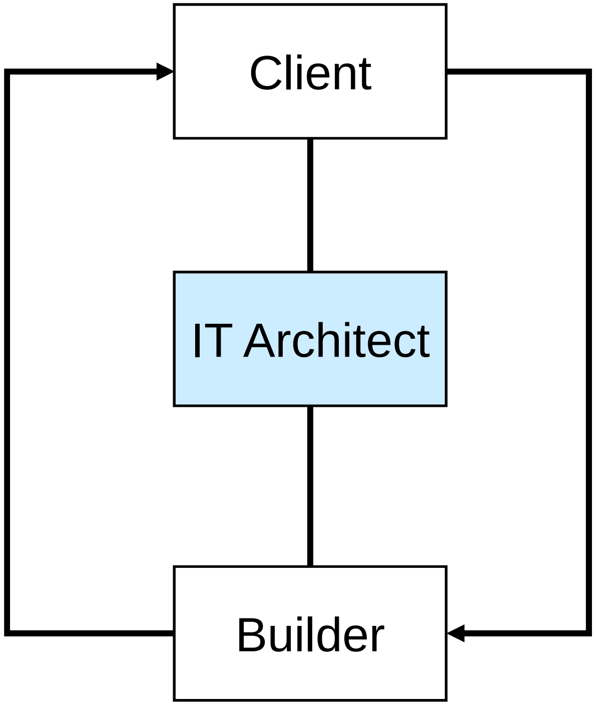
Operational stability

Decision making

Design integrity

Options: Builder Nominates, Customer decides
Prioritizing: Trade-offs, Options, and Choices

- During initial **requirements** gathering, the architect establishes important **baselines**.
- Unstated **expectations** for the design must be identified and validated.
- Requirements may need to be **altered, added, or deleted** to deliver an optimal solution.
- The choice between architectures may well depend upon which set of **drawbacks** the client can handle best.
- If **trade-off** results are inconclusive, then the wrong selection criteria were used. Find out [again] what the customer wants and why they want it, then repeat the trade-off using those factors as the [new] selection **criteria**.
- The architect must transcend the limitations of the builder's state of the art, and **imagine what is possible, given time and budget constraints**.



Architecting vs. Design vs. Engineering

- Architecting is
  - Working ***for*** a client and ***with*** a builder
  - Helping determine relative requirement **priorities**, acceptable performance, cost, and schedule
  - Taking into account such factors as technology **risk**, projected market size, likely competitive moves, economic trends, political regulatory requirements, project organization, and the appropriate "**ilities**" (availability, operability, maintainability, etc.)

Engineering is

- Working ***with*** an architect and ***for*** the builder
- Applying the best engineering practices to assure **compliance** at the system level with the designated architecture and with applicable specifications, standards, and contracts.

The essence of systems is relationships, interfaces, form, fit, and function.
The essence of **architecting** is structuring, simplification, compromise, and balance.
The challenge is control, if not the reduction of complexity and uncertainty.

The design of complex systems must blend the art of architecture with the science of engineering

The IT Systems Architect is also a Trusted Advisor
What clients expect

- Understand us, and like us
- Don't try to force things on us
- Give us **options**, increase our understanding of those options, give us their recommendations, and let us chose
- Help us think things through (it's our decision)
- Help us think and separate our logic from our emotion
- Give us **reasoning** (help us think), not just **conclusions**
- Help us to put our issues in context, through the use of metaphors, stories, and anecdotes (few problems are completely unique)
- Challenge our assumptions (help us uncover the false assumptions we've been working under)
- Criticize and correct us gently, lovingly
- We can rely on them to tell us the truth
- Are consistent (we can depend on them)

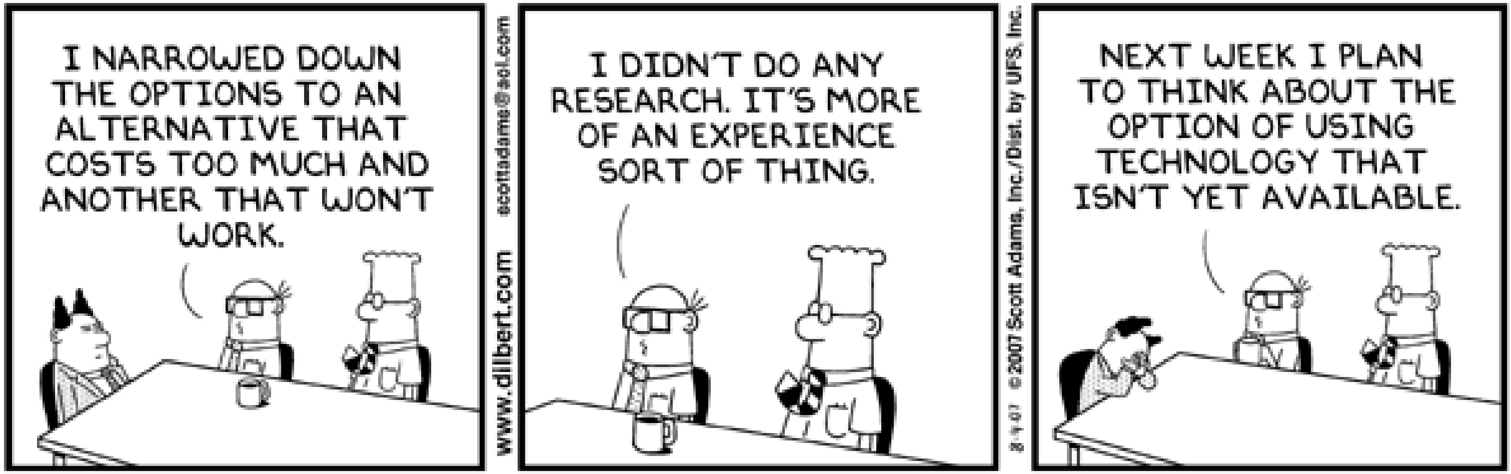
D. Maister, C. Green, R. Galford, The trusted Advisor, The Free Press, 2000.

Take a point of view (POV)

- It is useful to our clients if we articulate a Point Of View, even if it ends up being rejected or wrong.
- Two reasons:
  - It stimulates reactions
  - It crystallizes issues
- Stating a POV serves as a catalyst, a way of helping the client think
- Learn to express a POV with a simple, phrase such as:
  - Now let me just float a trial balloon here
  - Hey, who knows where this might go, but it occurs to me that ...

D. Maister, C. Green, R. Galford, The trusted Advisor, The Free Press, 2000.

Trusted Advisor and Architect Antipatterns


I NARROWED DOWN THE OPTIONS TO AN ALTERNATIVE THAT COSTS TOO MUCH AND ANOTHER THAT WON'T WORK.
scottadams@aol.com
www.dilbert.com
I DIDN'T DO ANY RESEARCH. IT'S MORE OF AN EXPERIENCE SORT OF THING.
©2007 Scott Adams, Inc./Dist. by UFS, Inc.
8-4-07
NEXT WEEK I PLAN TO THINK ABOUT THE OPTION OF USING TECHNOLOGY THAT ISN'T YET AVAILABLE.

© Scott Adams, Inc./Dist. by UFS, Inc.

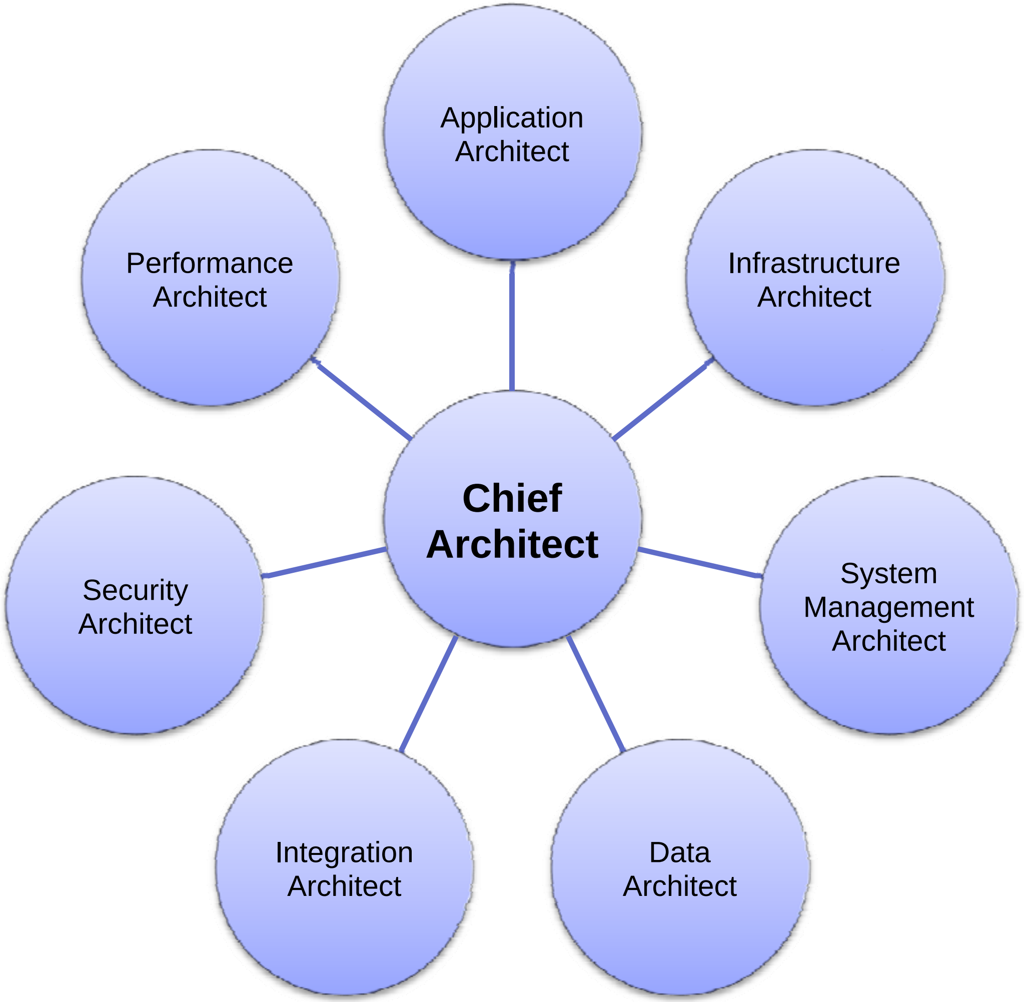
- Respond to the minutiae of an RFP rather than aggressively work relationships to manage the bigger picture with clients.
  - Risk of missing the big picture, leaving clients unconvinced that the architect's company has the capability to do anything other than implementation.
- Give the client a car when they need a bike
  - Over-solutioning and building up complexity are a sure way to failure

What are IT Architects and what do they do all day ?

IT Architect Roles and Responsibilities

Dr. Marcel Schlatter
IBM Distinguished Engineer
Member of the IBM Academy of Technology
marcel.schlatter@ch.ibm.com

Architect Roles



Congratulations!

- You have been chosen as the Chief Architect for a complex project. That means that you
  - Must correctly identify the problem to be solved
  - Oversee requirements development for the identified problem
  - Oversee development of a solution based on the requirements
  - Work closely with the project managers to obtain resources, time and budget to complete the project

- Your decisions
  - Have a direct relationship on the performance and viability of the solution
  - Have tremendous influence over the cost and delivery of the solution

Beware!

- You have been chosen as the Chief Architect for a complex project. That means that you
  - **Must correctly identify the problem to be solved**
  - Oversee requirements development for the identified problem
  - Oversee development of a solution based on the requirements
  - Work closely with the project managers to obtain resources, time and budget to complete the project

- Your decisions
  - Have a direct relationship on the performance and viability of the solution
  - Have tremendous influence over the cost and delivery of the solution

System Architecting Hall of Shame Candidates

- OS/2
  - IBM solved the **wrong problem** — needed to produce an inexpensive product with easy to use features, not a well tested, complicated consumer product
  - **Result:** Relegated to IT oblivion
- Beta
  - Sony solved the **wrong problem** — needed an inexpensive, flexible product with widespread licensing
  - **Result:** Loss of entire market category
- Napoleon's Grande Armée
  - Napoleon solved the **wrong problem** and equipped his army to fight the Russians, not typhus
  - **Result:** As we might say, the rest is history. System engineering principles don't just apply to IT systems
- Denver International Airport
  - Solution that was **too complicated**, not maintainable and not implementable, and had no backup solution
  - **Result:** Nearly 2 year delay in opening a $5 billion airport

Solving the Correct Problem

- Cannot over emphasize the need to solve the correct problem
- Solutions attempting to solve the wrong problem will
  - usually fail to meet to client’s needs
  - most of the time cost more
  - often prove impossible
- No amount of clever design, use of architectural patterns or use of sophisticated technology can overcome bad concept selection
- Need to understand the client’s **Wants and Needs**, in addition to **explicitly specified Requirements** as part of identifying, and then solving the correct problem

The Chief Architect Role is a Leadership Role

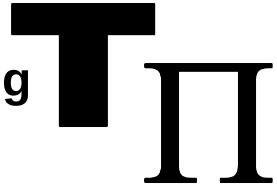
> *Management is doing things right;*
> *leadership is doing the right things.*
>
> Peter F. Drucker

Chief Architect Roles and Responsibilities

- Provide the **technical leadership** necessary to implement or achieve a business strategy through an IT solution
- Carry end-to-end **technical solution responsibility**
- Carry the whole scope of the **problem to be solved**, and the **solution** in his/her head
- Technical management of **Requirements, Issues, Risks & Changes**
- Definition of applicable **Architectural Principles**
- Manage **reviews**
  - Work products and deliverables
  - Co-ordinating external reviewers, Quality Assurance
- **Internal**: Advise the program manager and project executive on all aspects of the technical solution
- **External**: Develop relationships with client technical executives

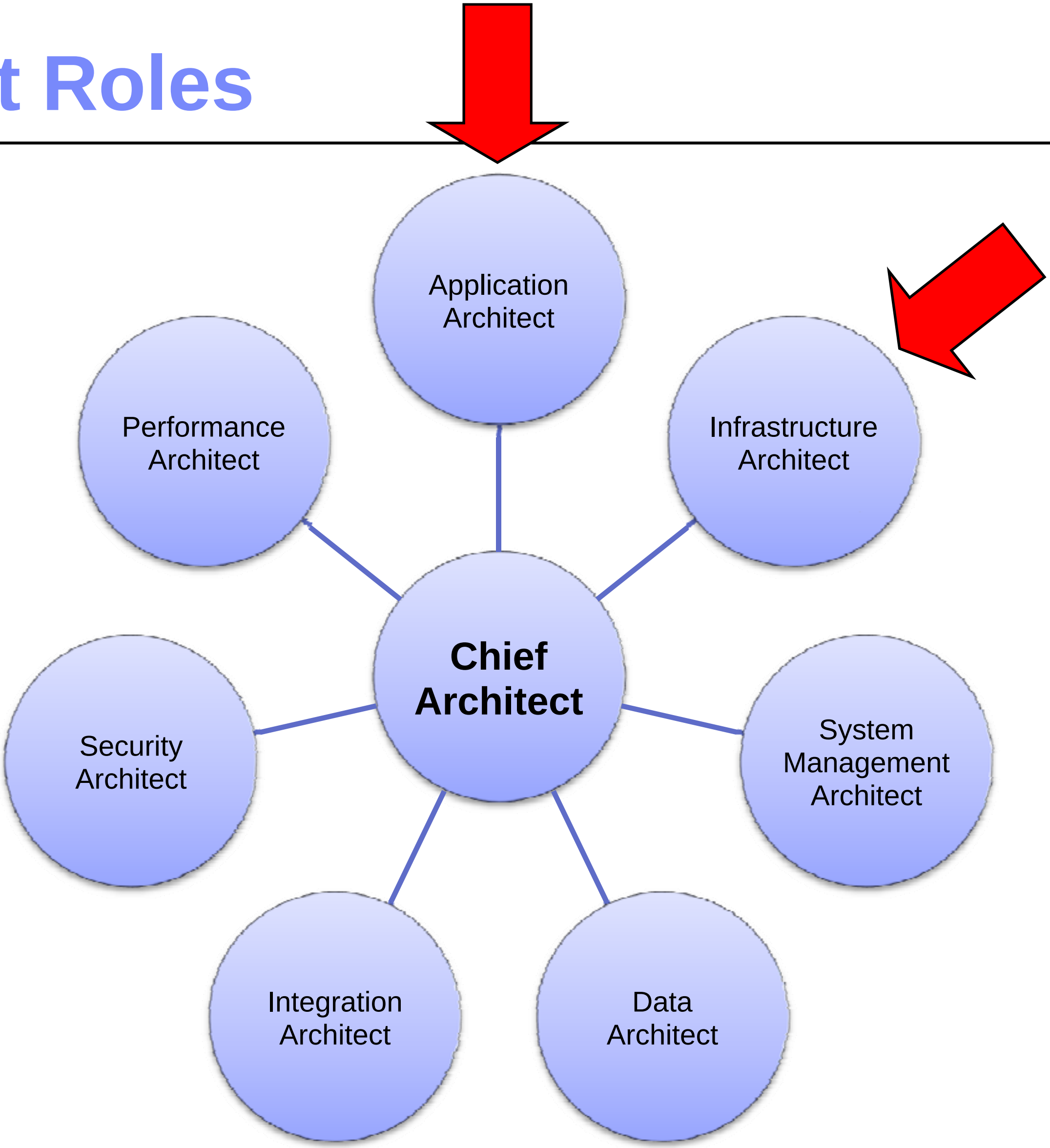
What Your Mother Would Tell You About Being a Chief Architect

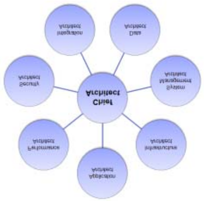
- Lead development of requirements, solutions, plans and all manner of contingencies, but let your **team** work the details
- As Chief Architect, you are not the chief technologist
  - You need to understand, but not necessarily be an expert in all aspects of the technology
- **Know something about everything, but everything about nothing**



- Learn how to identify the client’s key **requirements**
- Build a system using technology to **solve the client’s problems**. If you build because of technology, you will be creating a problem
- Spend some extra time making sure you can **identify the correct problem** to solve
- Don’t be afraid to change a solution if it won’t work
- Use more than your intuition to evaluate solutions
- Don’t be afraid to **build on other people’s solutions**
- Know how to **optimize** and **estimate**

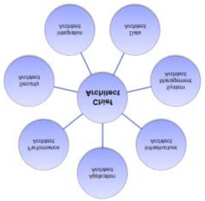
Architect Roles





Application Architect

- Defines **what the solution does**
- Responsible for the **Functional Aspects** of the system
- Key responsibilities
  - Understands how the business requirements can be met using application software, and defines what **application software packages** and / or **bespoke code** is needed
  - Develops and maintains **application architectures** and strategies and to ensure the design integrity of the application subsystem and that it meets the agreed requirements
  - Defines **high level data flows** between applications
  - Leads any **bespoke application development**
  - Leads the **configuration of the application software**



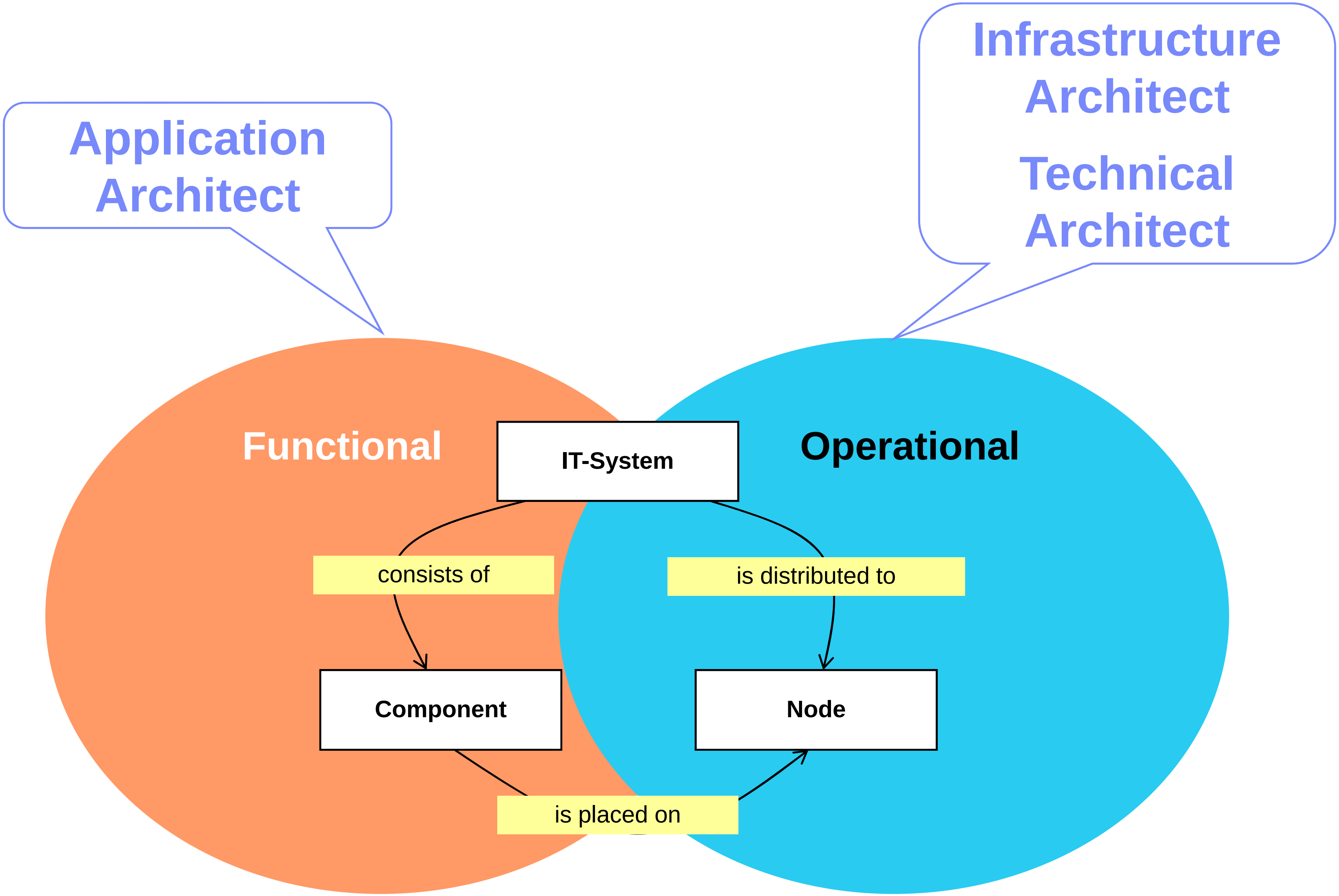
Infrastructure (or Technical) Architect

- Defines the overall system shape
  - What the **building blocks** are from which the solution will be made
  - How the **data** and **functionality** will be **placed**

- Responsible for the **Operational Aspects** of the system

- Key responsibilities
  - Establishes **non-functional and technical infrastructure requirements**
  - Defines the **infrastructure** solution
    - Networking, hardware configurations, system software, middleware
  - Performance, Capacity, Scalability
  - Availability, Recoverability
  - Systems Management, Service Levels

Non-Functional Requirements


Application Architect
Infrastructure Architect
Technical Architect
Functional
Operational
IT-System
consists of
is distributed to
Component
Node
is placed on

The **Application Architect** is responsible for the Functional Aspects, which include these key concepts:

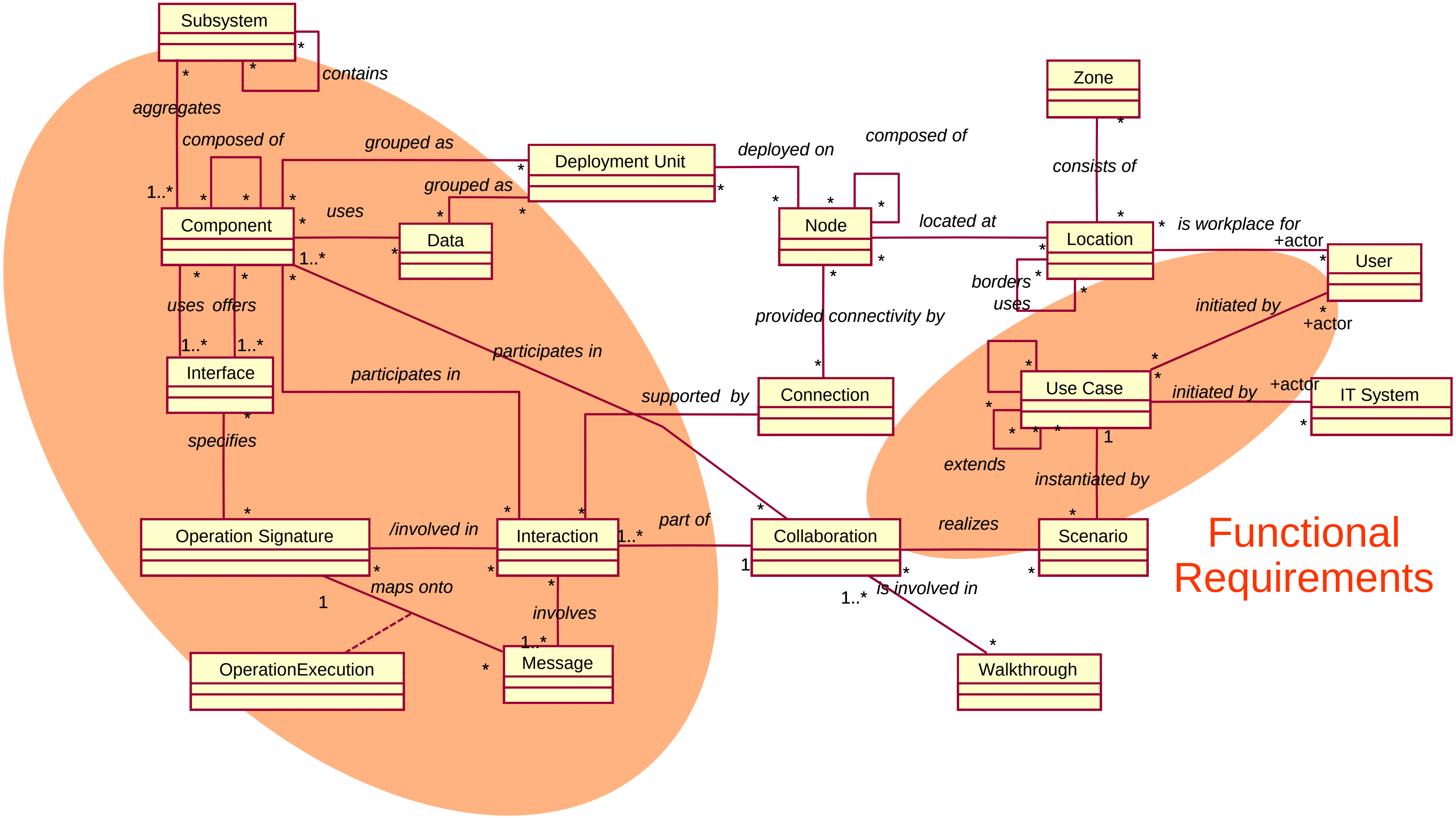
- **Component**
  - Modular unit of functionality which makes this functionality available through an interface
- **Subsystem**
  - Any grouping of components in IT system
- **Interaction and Collaboration**
  - Collaboration between components
  - Sequence of component operations
  - Exchanges between two components
  - Interface usage contract / protocol

Link between Use Cases, and Components

Use Case Realizations

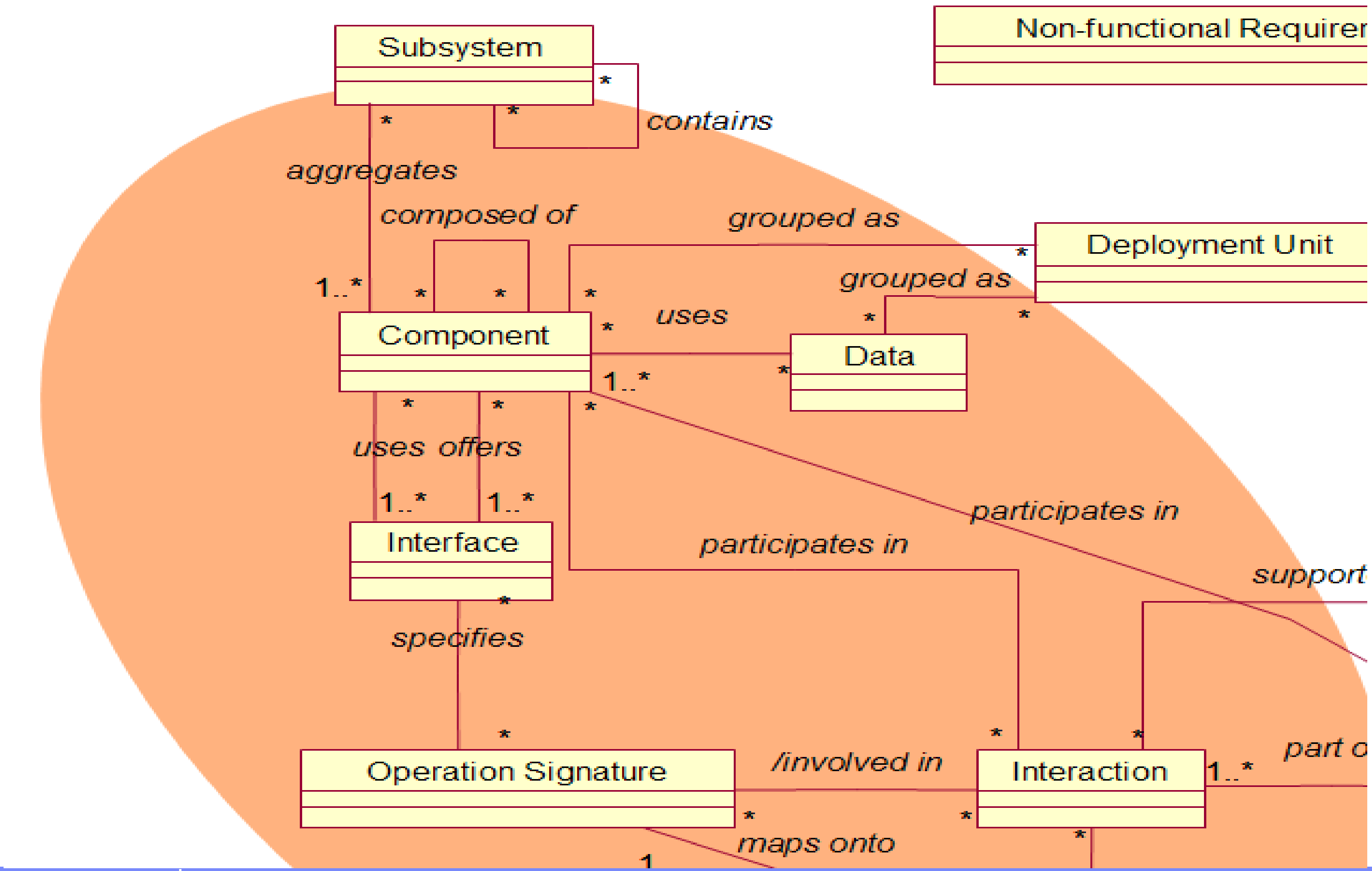
- **Data**

The Application Architect is responsible for the Functional Aspects, represented by the concepts in the highlighted areas below

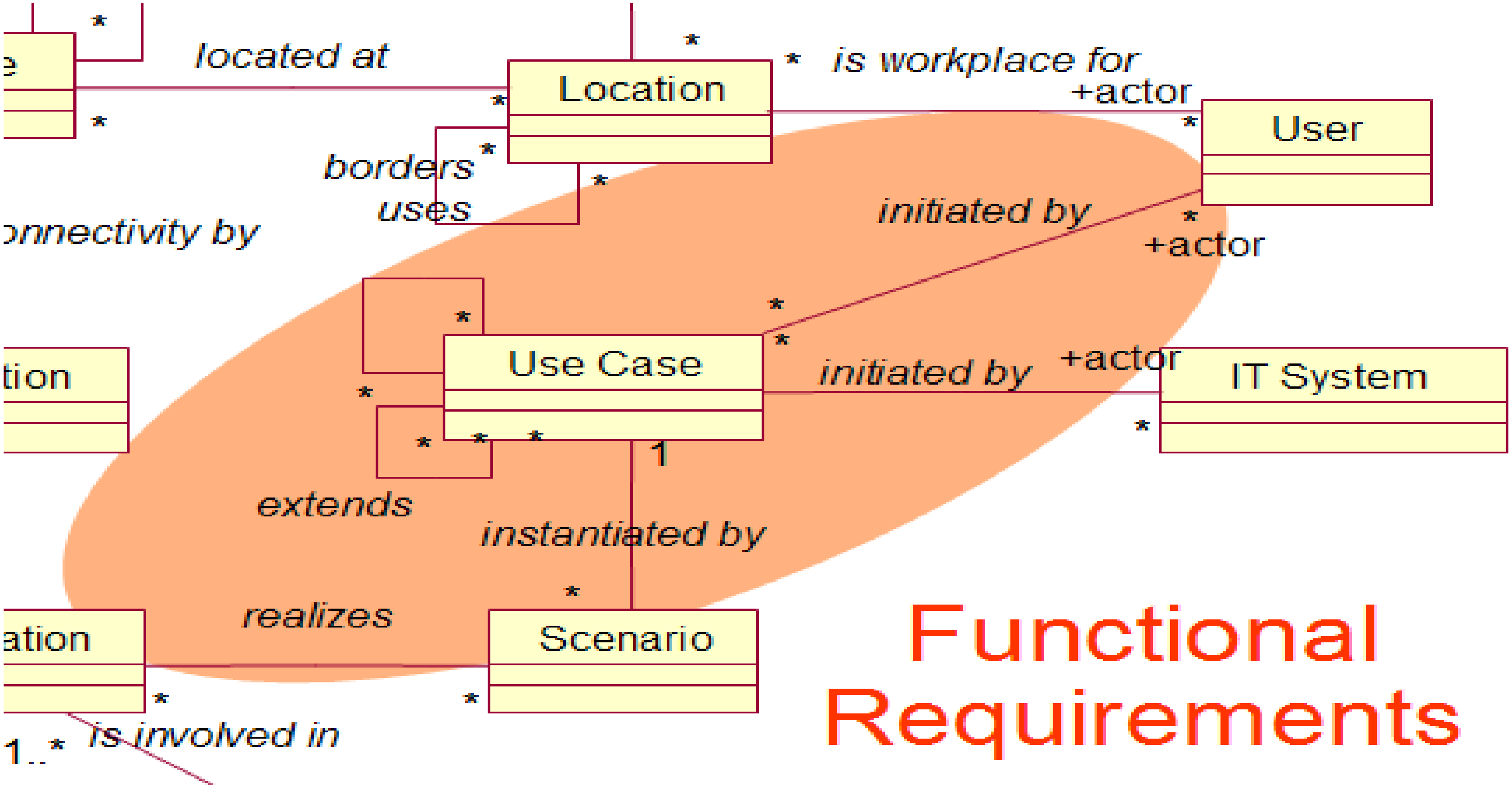


Functional Requirements

The Application Architect is responsible for the Functional Aspects, represented by the concepts in the highlighted areas below



The Application Architect is responsible for the Functional Aspects, represented by the concepts in the highlighted areas below

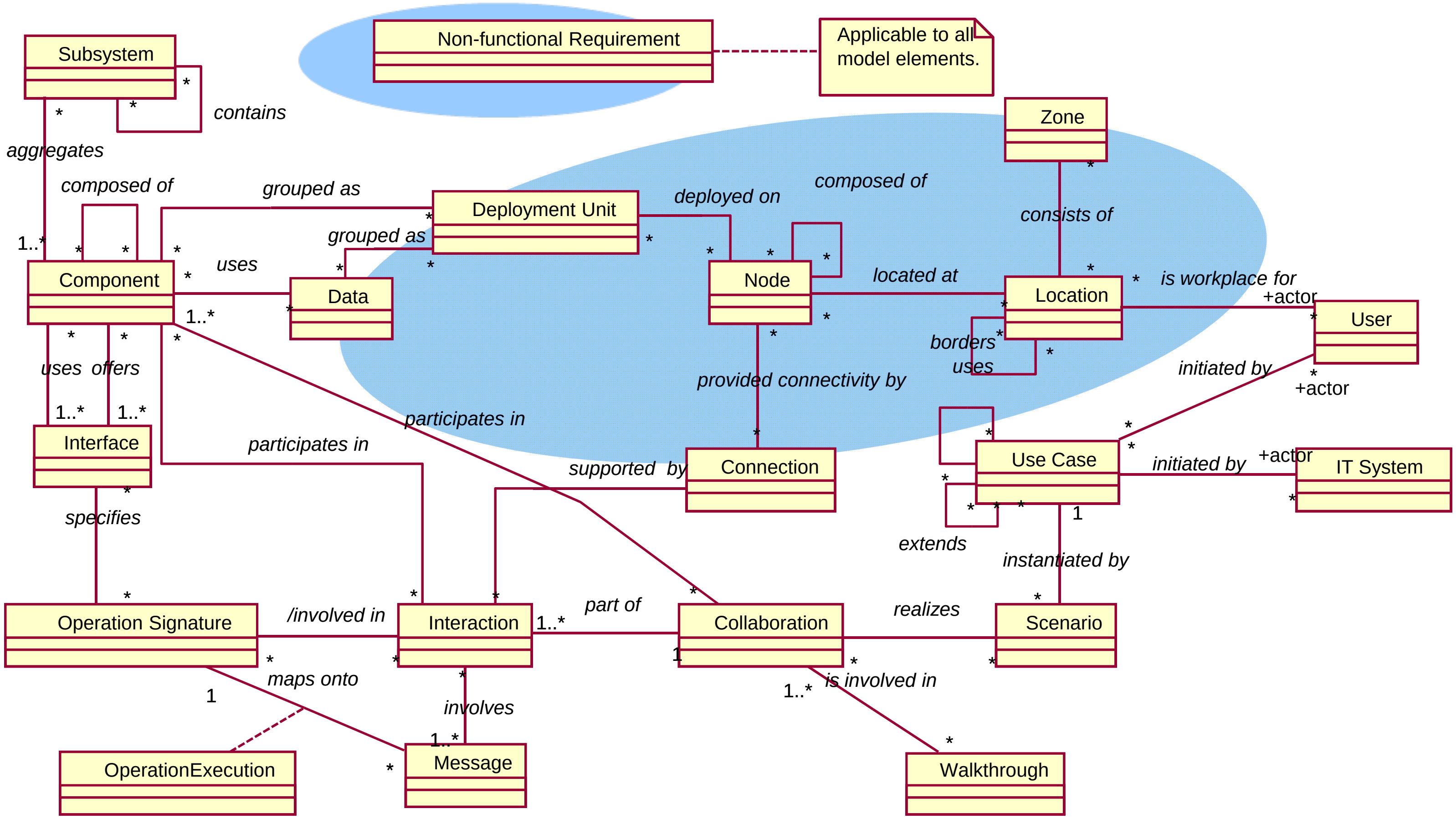


Functional Requirements

The Infrastructure Architect is responsible for the Operational Aspects, which include these key concepts:

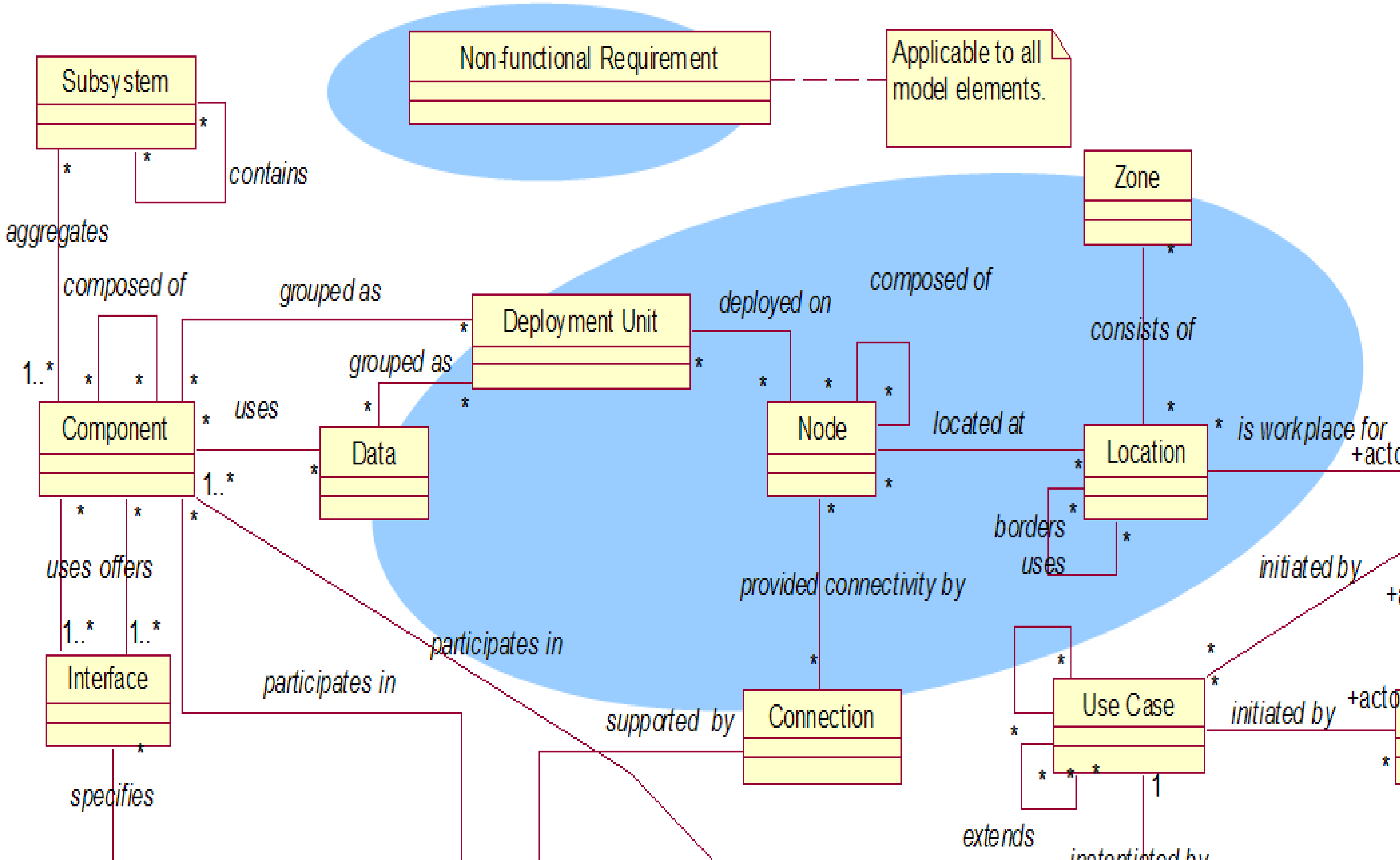
- **Node**
  - platform on which software executes
- **Location**
  - type of geographical area or position
- **Zone**
  - an area for which a common set of non-functional requirements can be defined
- **Connection**
  - physical data path between nodes (LAN, WAN, dial-up etc)
- **Deployment Unit**
  - one or more components placed together on a node
- **Non-functional Requirements (NFRs)**
  - Service Level Requirement (SLR) like performance, availability, etc.
  - Constraints: business / geography, IT Standards, current Infrastructure, etc.
- **Walkthrough**
  - description of the flow of a scenario starting from a user all the way through the system and back to the user

The Infrastructure Architect is responsible for the Operational Aspects, represented by the concepts in the highlighted areas below



Goal: Fulfill Non-functional Requirements (Service-Level, Manageability, etc.)

The **Infrastructure Architect** is responsible for the Operational Aspects, represented by the concepts in the highlighted areas below


Non-functional Requirement
Applicable to all model elements.
Subsystem
contains
aggregates
composed of
grouped as
Deployment Unit
deployed on
composed of
Zone
consists of
grouped as
Component
uses
Data
Node
located at
Location
is workplace for
+acto
borders
uses
initiated by
uses offers
provided connectivity by
participates in
participates in
Interface
supported by
Connection
Use Case
initiated by
+acto
specifies
extends
instantiated by
1..*
1
*