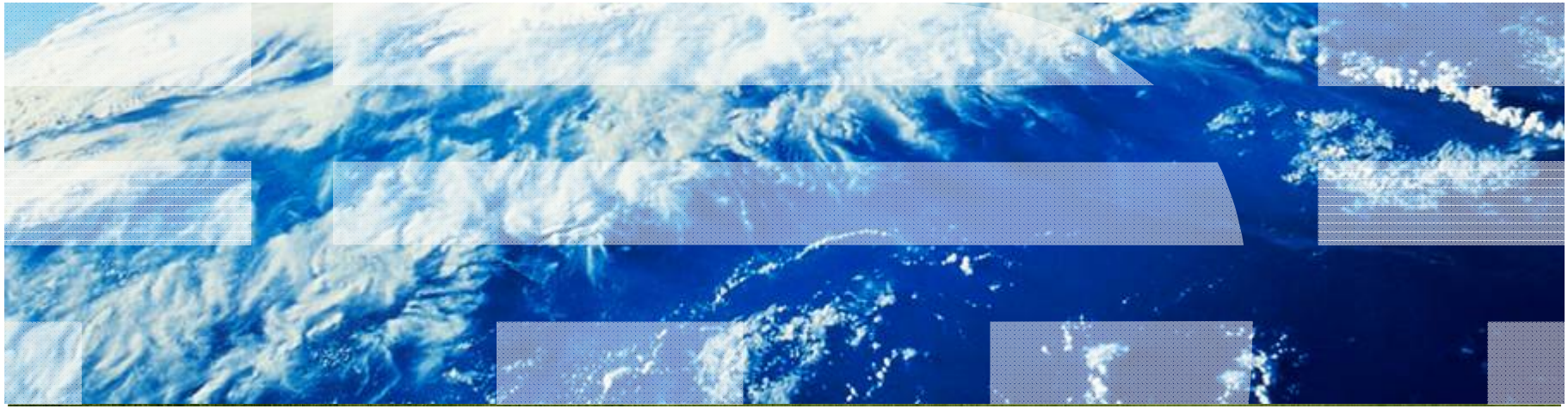
Dr. Marcel Schlatter
IBM Distinguished Engineer
Member of the IBM Academy of Technology

University of Zurich IBM

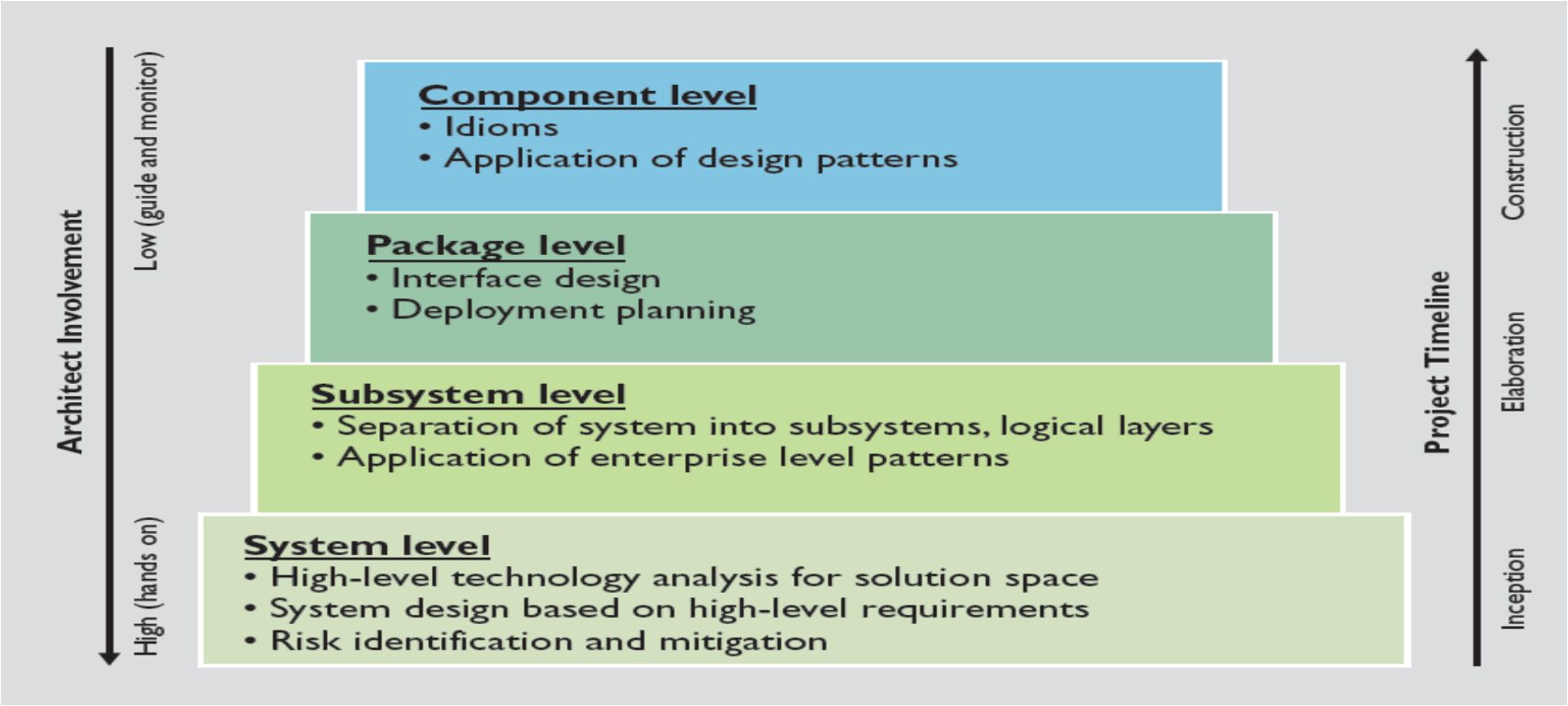
Enterprise IT Architectures

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

IT Architect Roles and Responsibilities



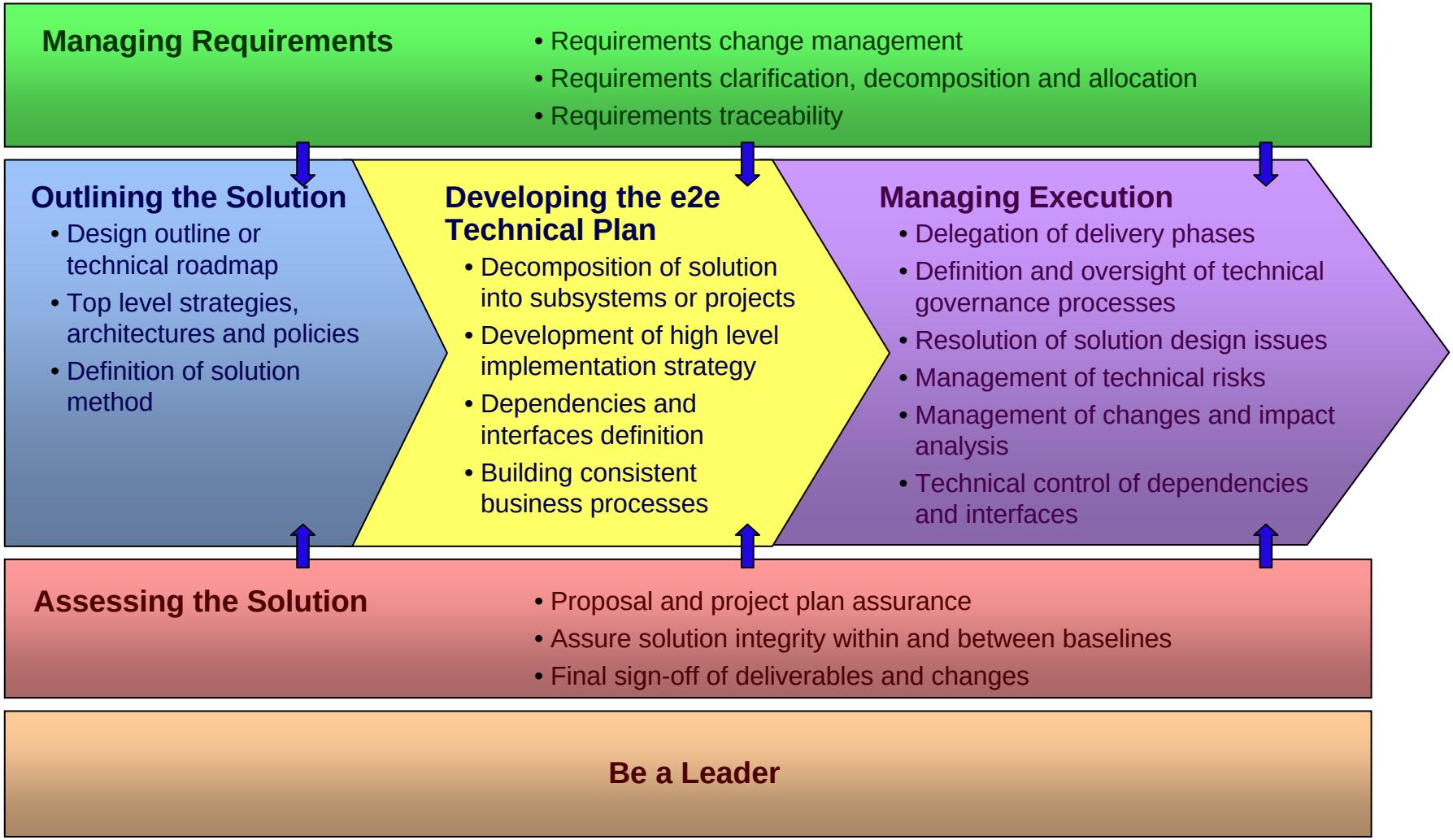
IT System or Solution Architects are technically competent system-level thinkers, guiding planned and economically efficient design processes to bring a system into existence.



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



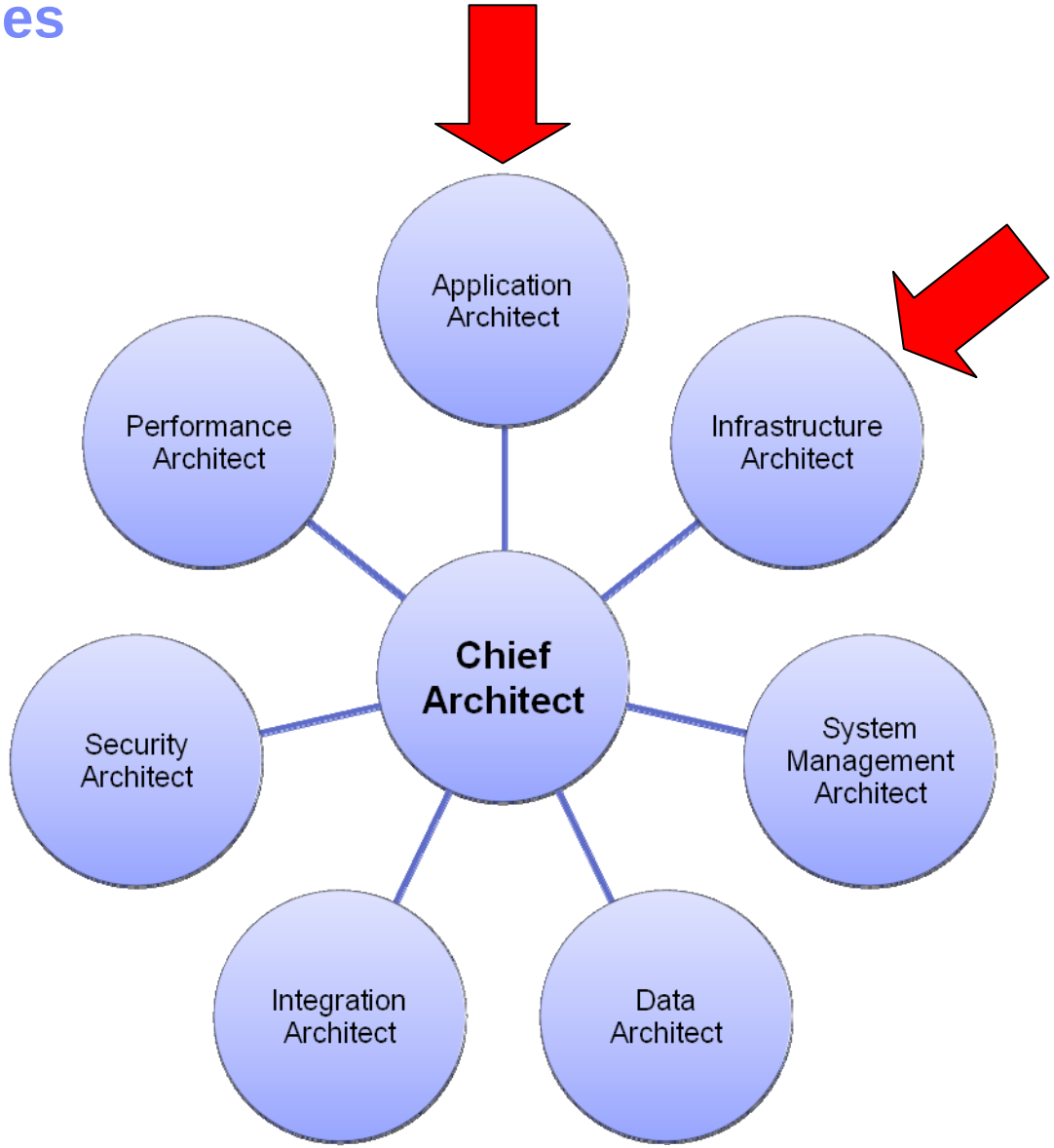
Architect Roles

Architect Roles

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**

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

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 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**

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