Software Architecture

Definition

Gabriel Pedraza

pedraza@uis.edu.co

Reminder: two levels of design

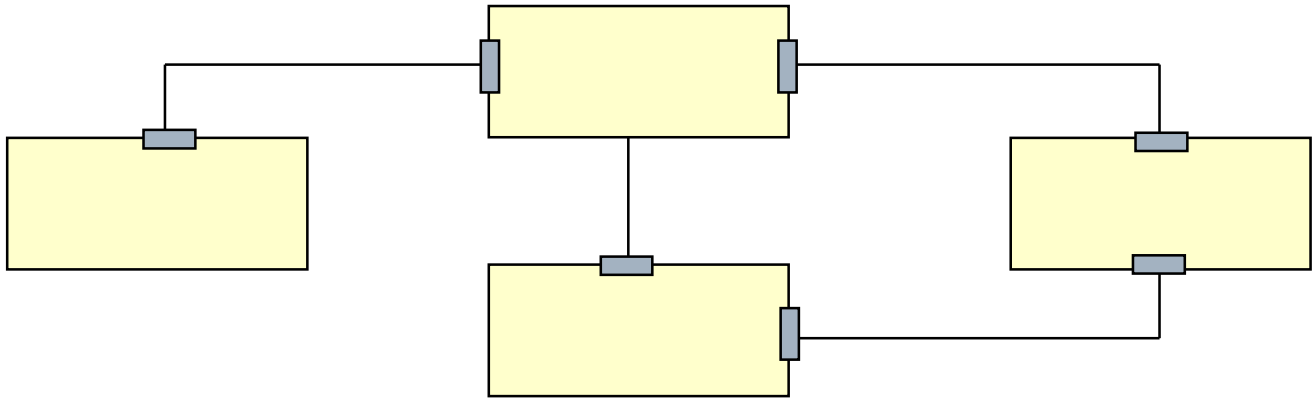
- Global design
  - Software architecture
    - Components
    - Connectors
- Detailed design
  - Programming concepts
    - Objects, structured types, …
    - Methods, procedures, …

Outline

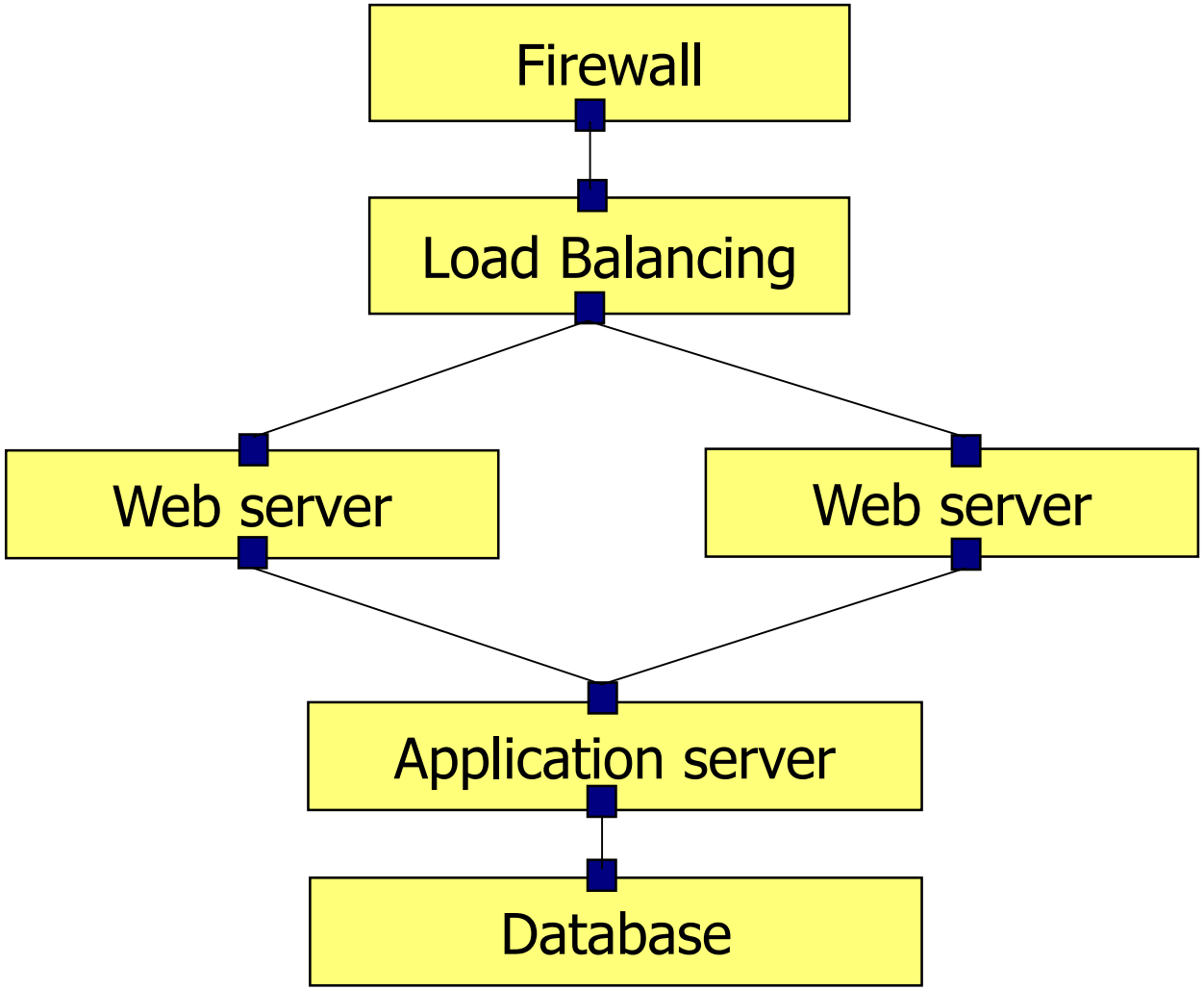
- Definition
- An important activity
- A difficult activity
- Conclusion

Software architecture

- Abstract specification expressed as components interacting through connectors
  - Its purpose is not to be executable
  - It is a decomposition that has to meet requirements at hand



Example

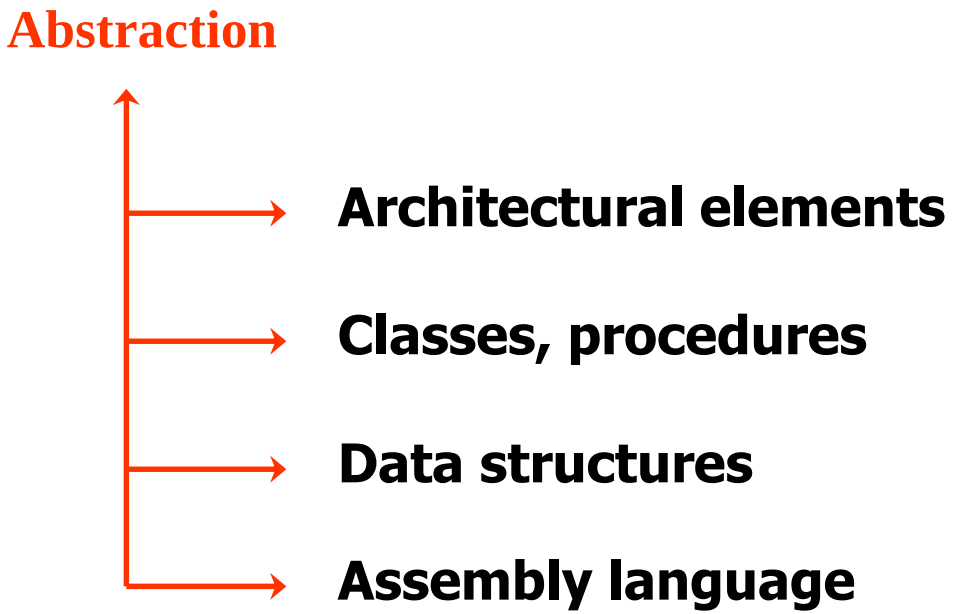

Firewall
Load Balancing
Web server
Web server
Application server
Database
Legend
C/S

Software architecture - specification

- An architecture defines the overall organization of a system
  - It is a guideline for designers / developers
- It must be precise
  - Choices are made
  - Not a (fuzzy) functional decomposition
- It is complete
  - Gives all necessary information for detailed design
    - Possibly performed by partners, remote teams, …

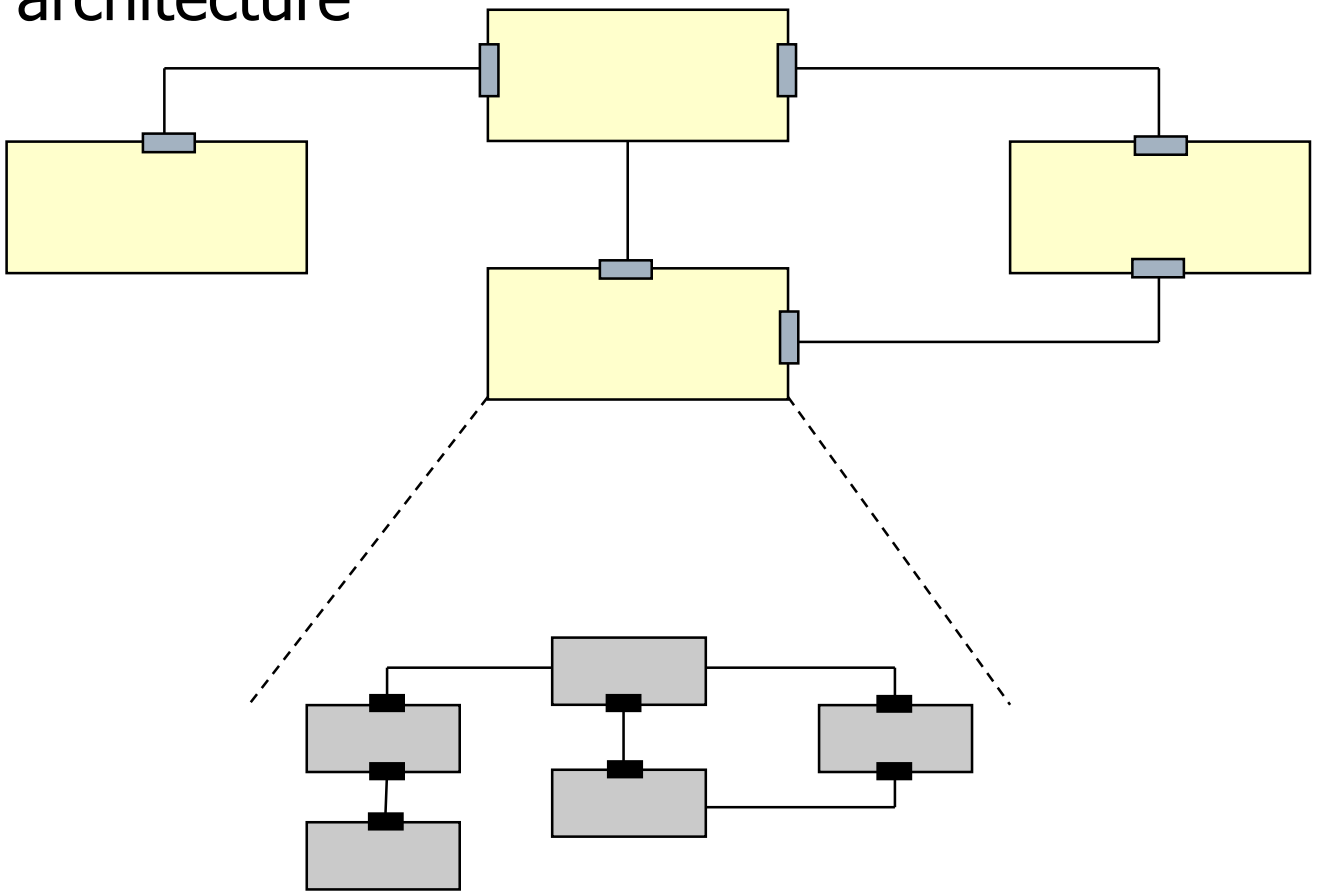
Software architecture - abstraction

- An additional level of abstraction
  - It does not bother with implementation details
  - It defines relevant properties of structuring elements



Software architecture - abstractions

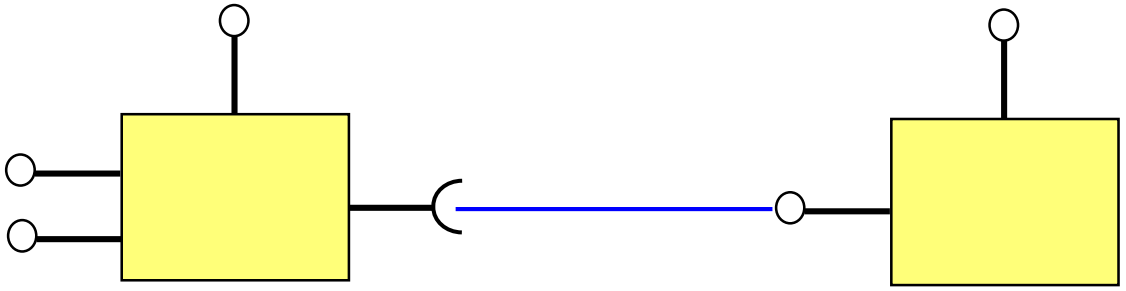
Global architecture



Architecture of an element

Software architecture - component

- Functional unit specification
  - Functional and non functional interfaces
  - Dependencies
  - properties, constraints
  - high cohesion



Software architecture - connectors

- First order objects
  - Defines interactions between components
  - Can be pretty complex
  - No standard specification language yet
  - Low coupling

Several kinds of architecture

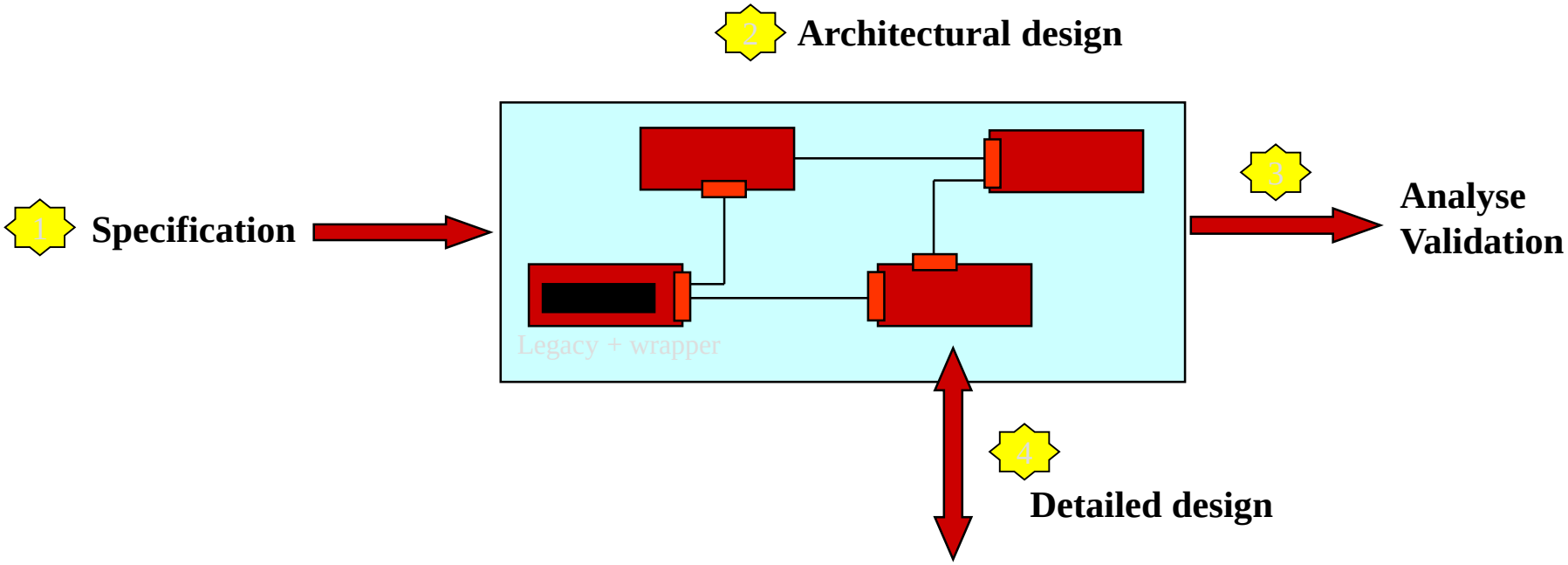
- Functional architecture (DAF)
- Software architecture (DAL)
- Technical architecture (DAT)

Outline

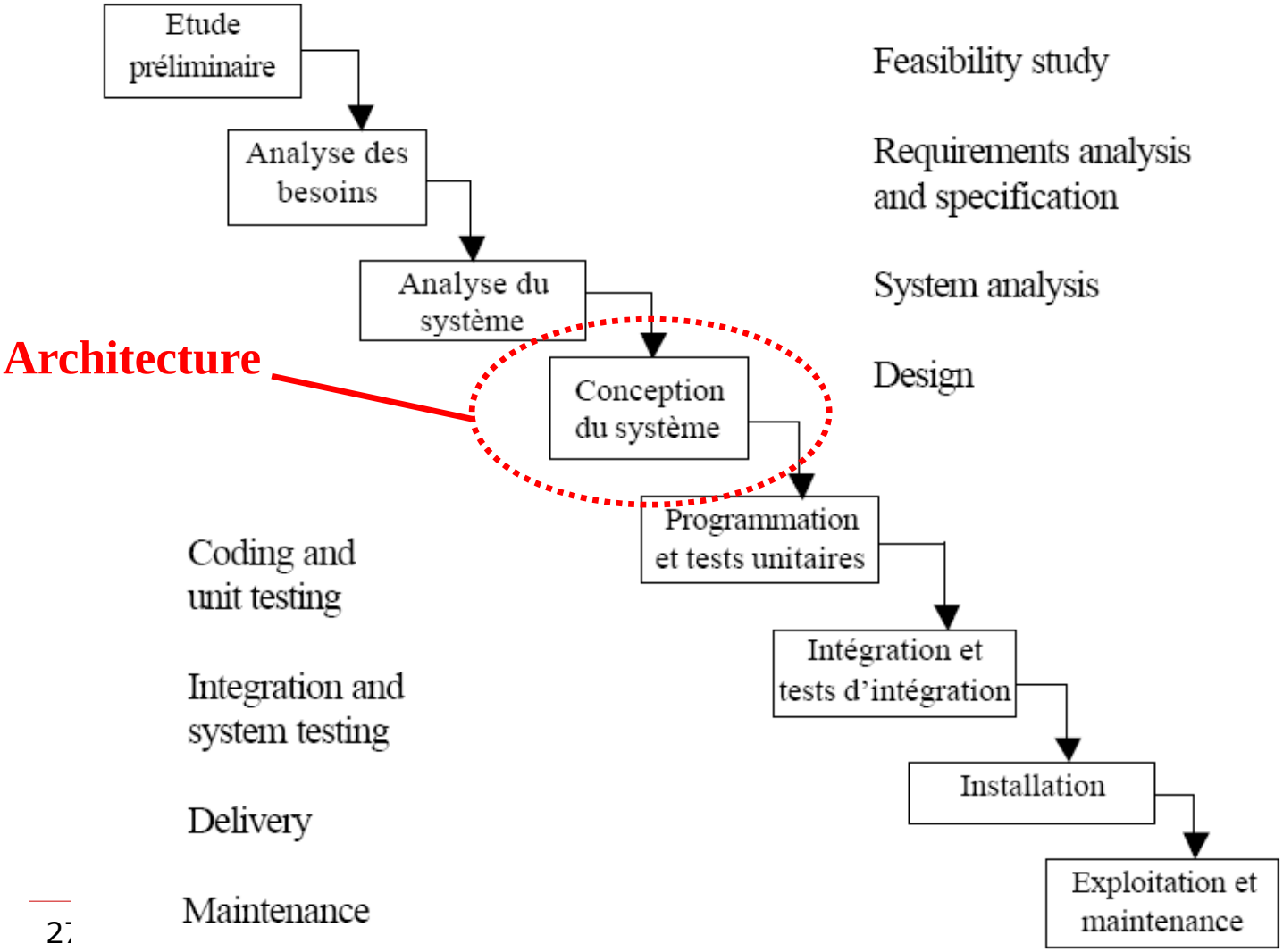
- Definition
- Architecture and lifecycle
- An important activity
- A difficult activity
- Conclusion

Position

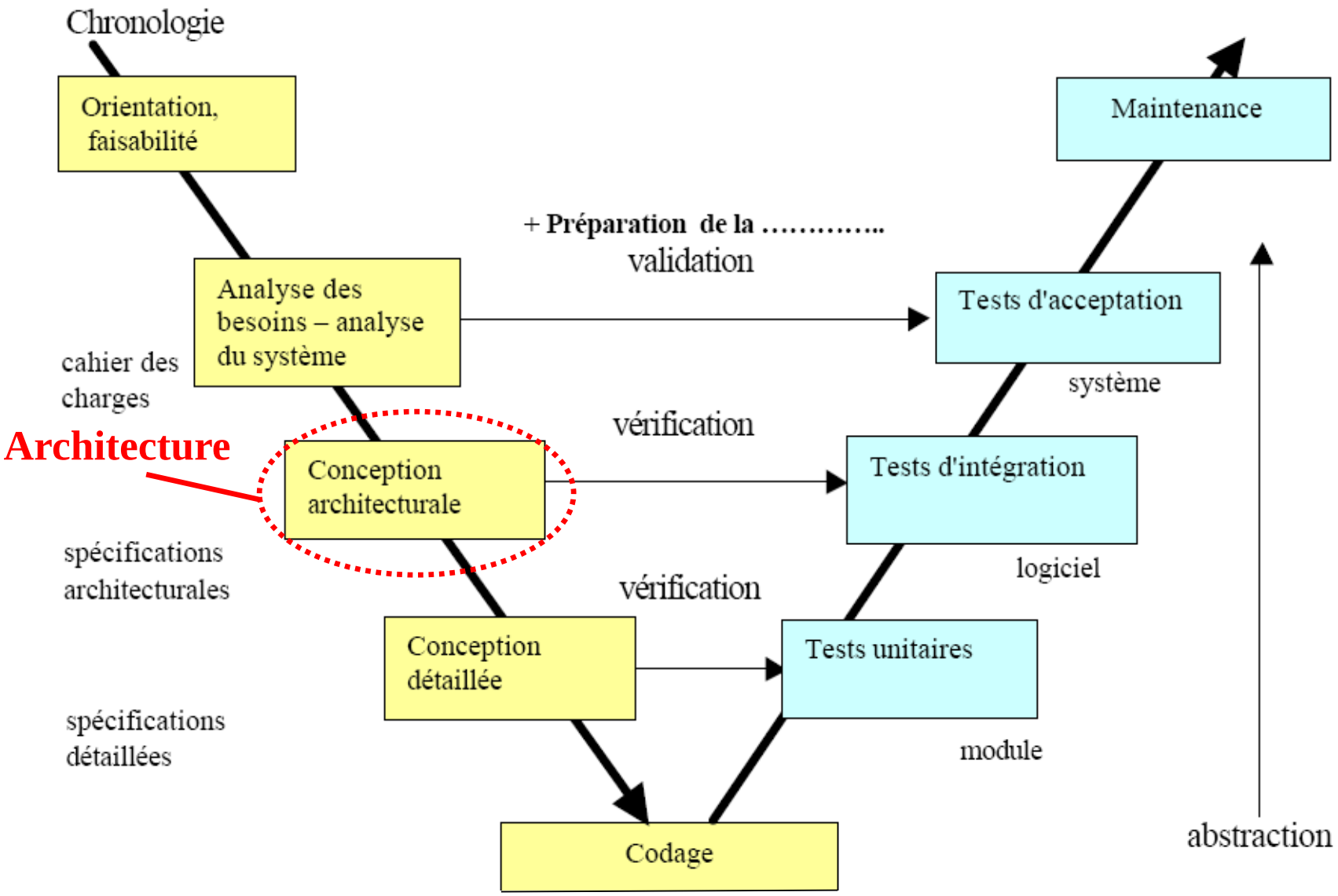
- First design step
  - Decompose the system into software components to reduce its complexity



Waterfall model

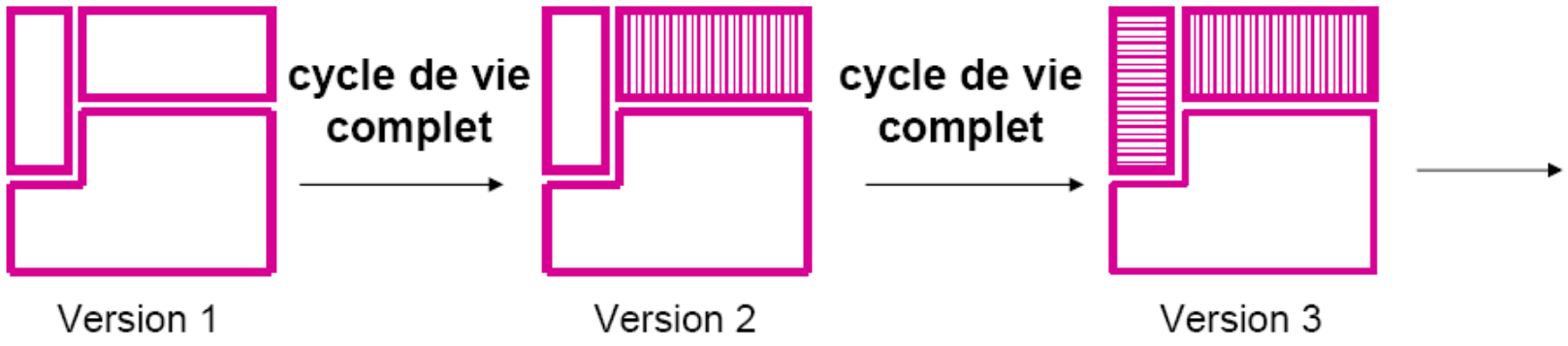


V model


Chronologie
Orientation, faisabilité
Maintenance
+ Préparation de la ……..
validation
Analyse des besoins – analyse du système
Tests d'acceptation
cahier des charges
système
vérification
Architecture
Conception architecturale
Tests d'intégration
spécifications architecturales
logiciel
vérification
Conception détaillée
Tests unitaires
spécifications détaillées
module
Codage
abstraction

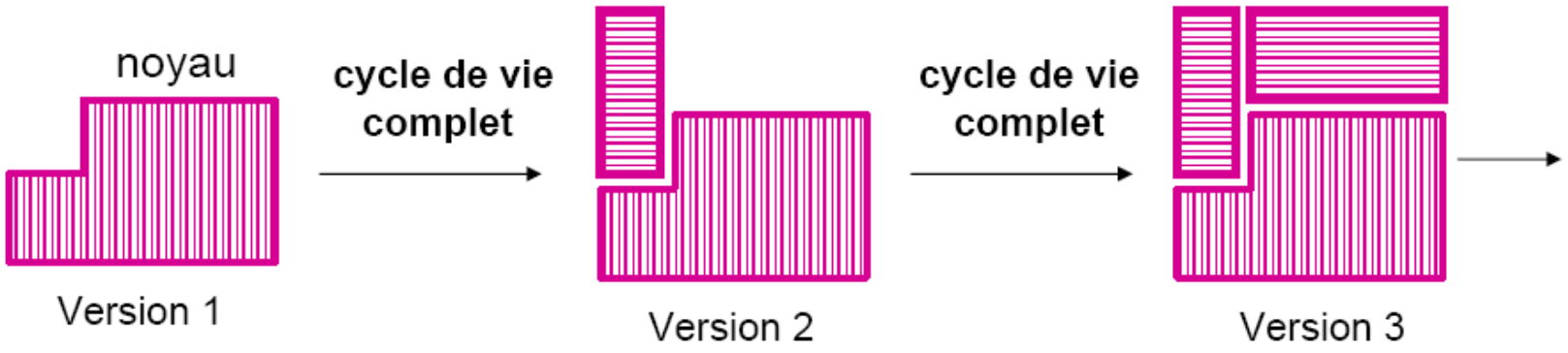
Incremental model - 1

- Stable architecture
  - The first increment defines the architecture
  - Next increments provide part of the architecture
  - Parallel development is possible



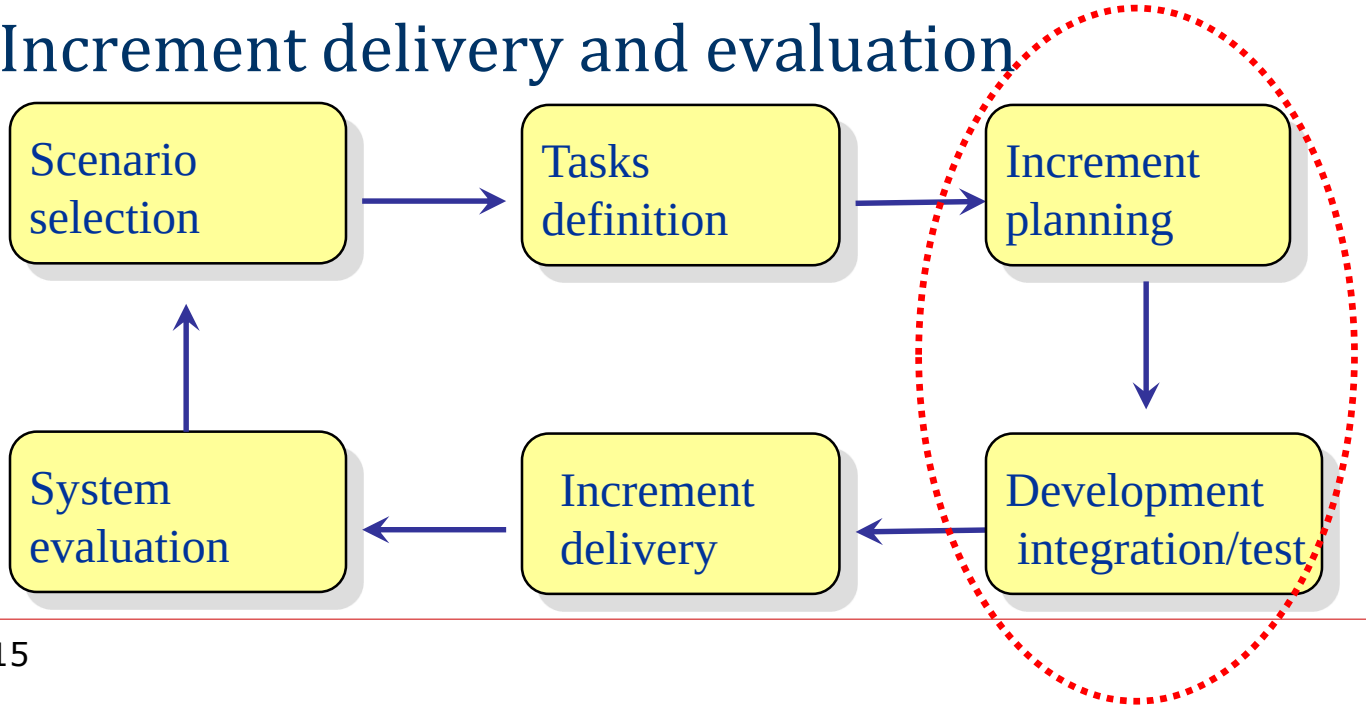
Incremental model - 2

- Evolving architecture
  - The first increment builds the core
  - Next increments build on top of the core
  - Each increment implements a complete lifecycle



Extreme programming (XP)

- Frequent iterations
  - Scenario selection (cards)
  - Tasks definition and distribution
  - Planning and tests
  - Increment delivery and evaluation



Conclusion

- Architecture explicitly appears in every software process
  - During design (and validation)
  - A very important position
    - Link between requirements management and detailed design
    - Half of the road has been made …

Outline

- Definition
- Architecture and lifecycle
- An important activity
- A difficult activity
- Conclusion

A key step

- Communication support
- Project structuration
- Technological choices
- Impact on quality and business

Communication vector

- Architecture provides a framework allowing most stakeholders to interact
  - Presentation
    - to clients, managers
  - Negotiation
    - about requirements, evolutions, …
  - Project management
    - Quality evaluation, progress assessment,
- Issue: sometimes, too many interactions !

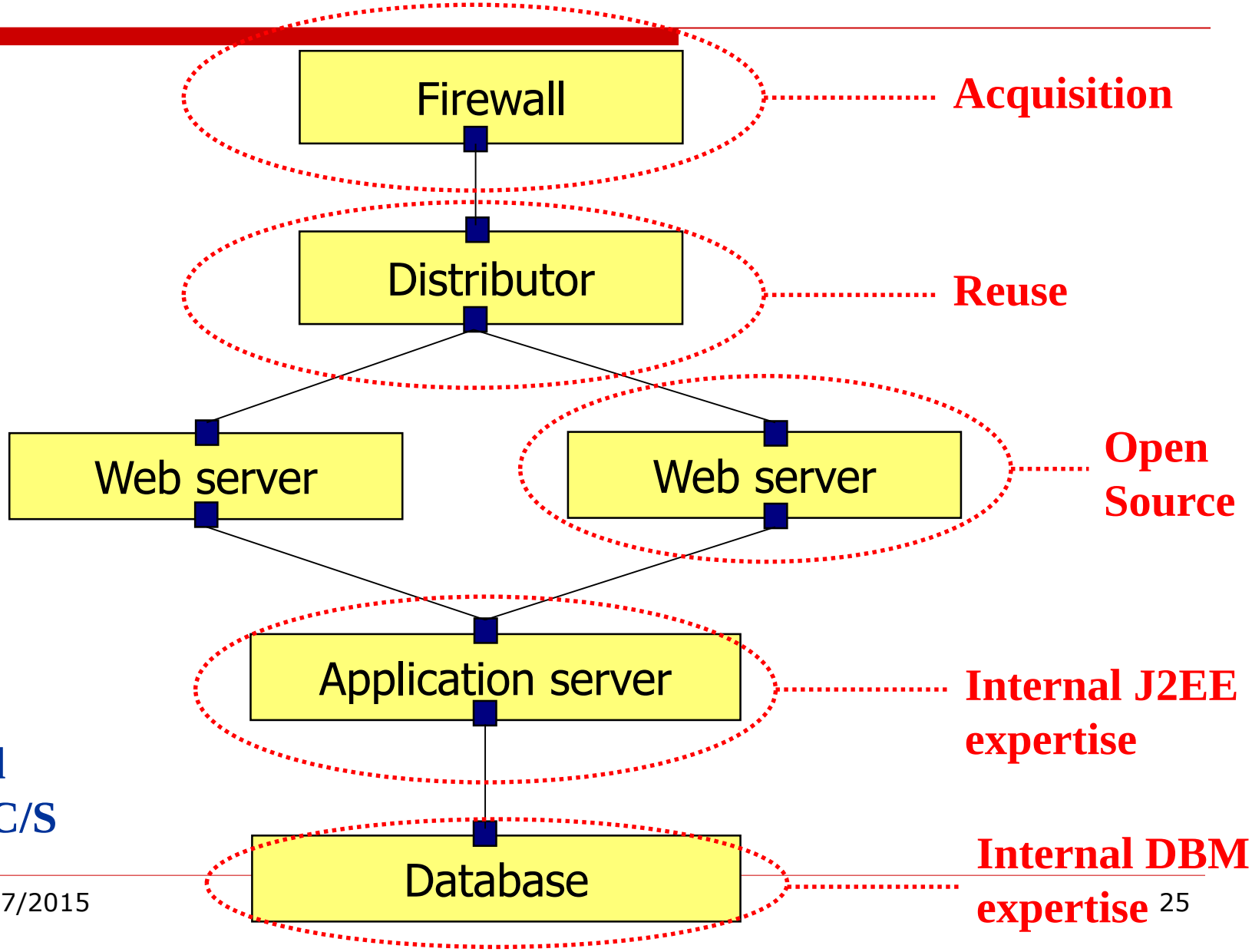
Project structuration

- Development plan definition
  - Tasks definition
  - Task allocation definition
  - Roles / duties definition
- Interaction definition
  - How teams are going to interact
  - Methods, documents to be produced, …

Technologies definition

- Technological choices
  - Technologies to be used
    - J2EE, .Net, …
  - Components to be developed
  - Components to be reused
  - Components to be bought
    - Components Off the Shelves (COTS)
  - Partnership definition
    - IBM, Microsoft support

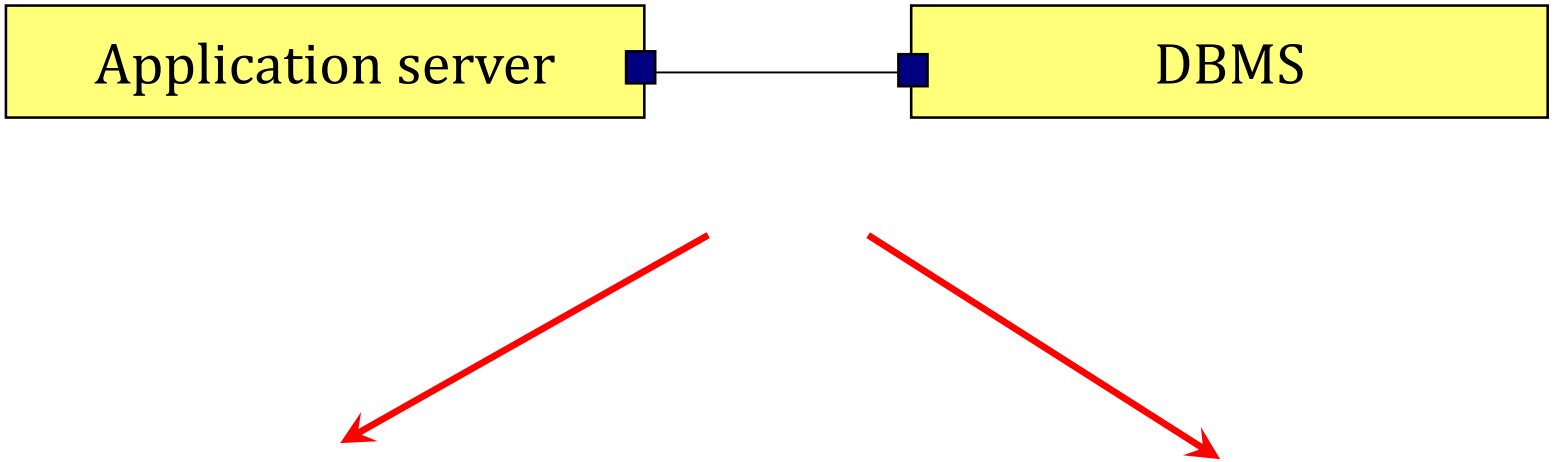
Example


Firewall
Distributor
Web server
Web server
Application server
Database
Acquisition
Reuse
Open Source
Internal J2EE expertise
Internal DBMS expertise
Legend
C/S

Impact on software quality

- Architecture has a strong impact on the system final properties
- An architecture favors or degrades non functional properties
  - Performance and efficiency
  - Security and safety
  - Reliability and availability, …
- A first compromise is made at the architectural level. In a definitive manner …

Example



On the same machine

+ reliability
+ performance

On two different machines

+ availability (caches, …)
+ maintainability

Example

- More examples
  - A small number of components favors performance but not maintainability
  - A large number of components favors maintenance but not performance
  - Redundancy favors safety but not security
- It is the beginning of design headaches …

Impact on business

- Deployment of an architecture allows
  - the creation of a common assets
    - Reusable components, specialized tools, training sessions
  - Various optimizations
    - Learning curve, infrastructure costs, …
- An enterprise architecture is an infrastructure shared by the whole company
  - Hard to change !

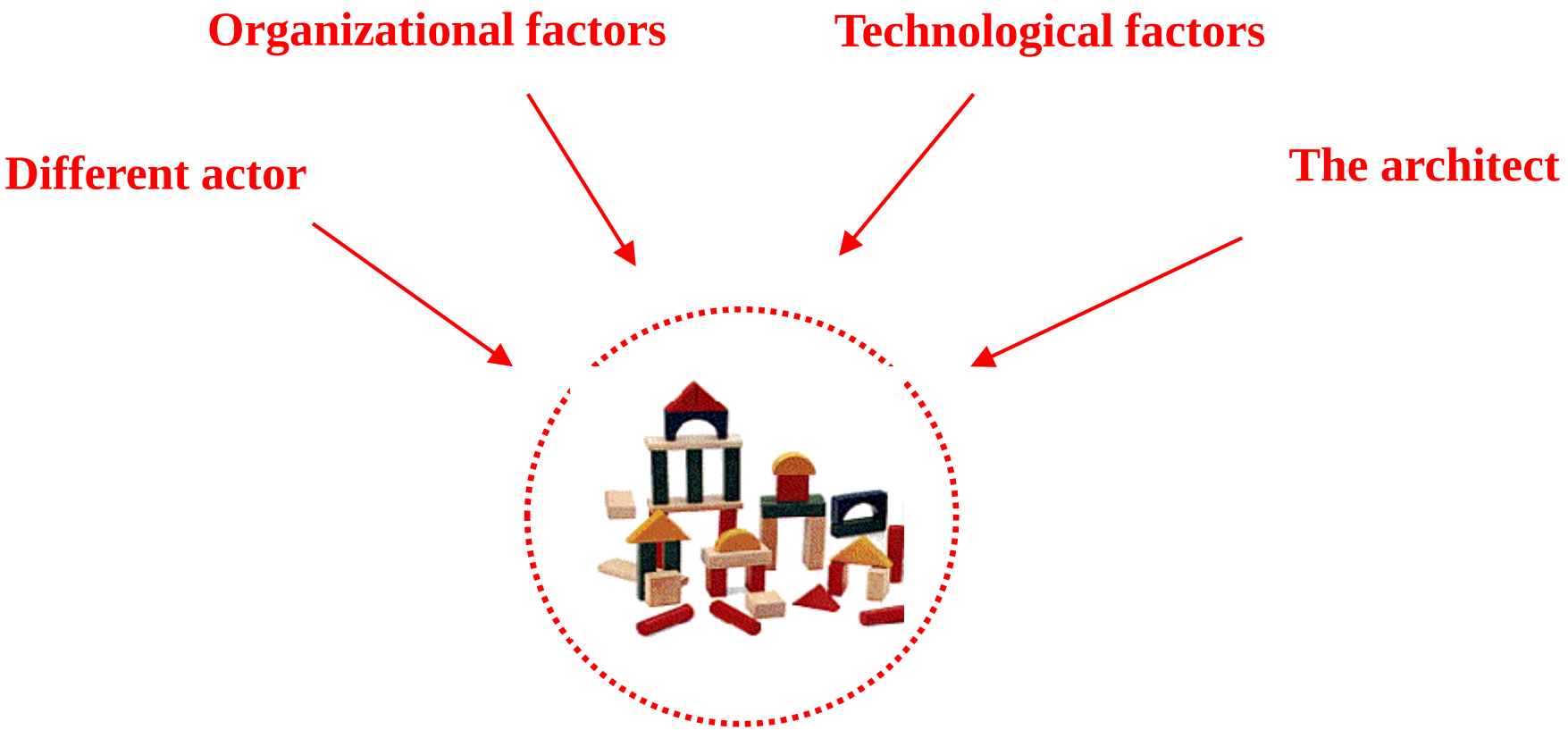
Conclusion

- Architectural decisions have an important, lasting impact
- Architectures must be carefully validated as soon as possible
  - Evaluation reviews
  - Prototypes development
  - Key technologies evaluation
- Architecture influences quality … but does not provide any guarantees

Outline

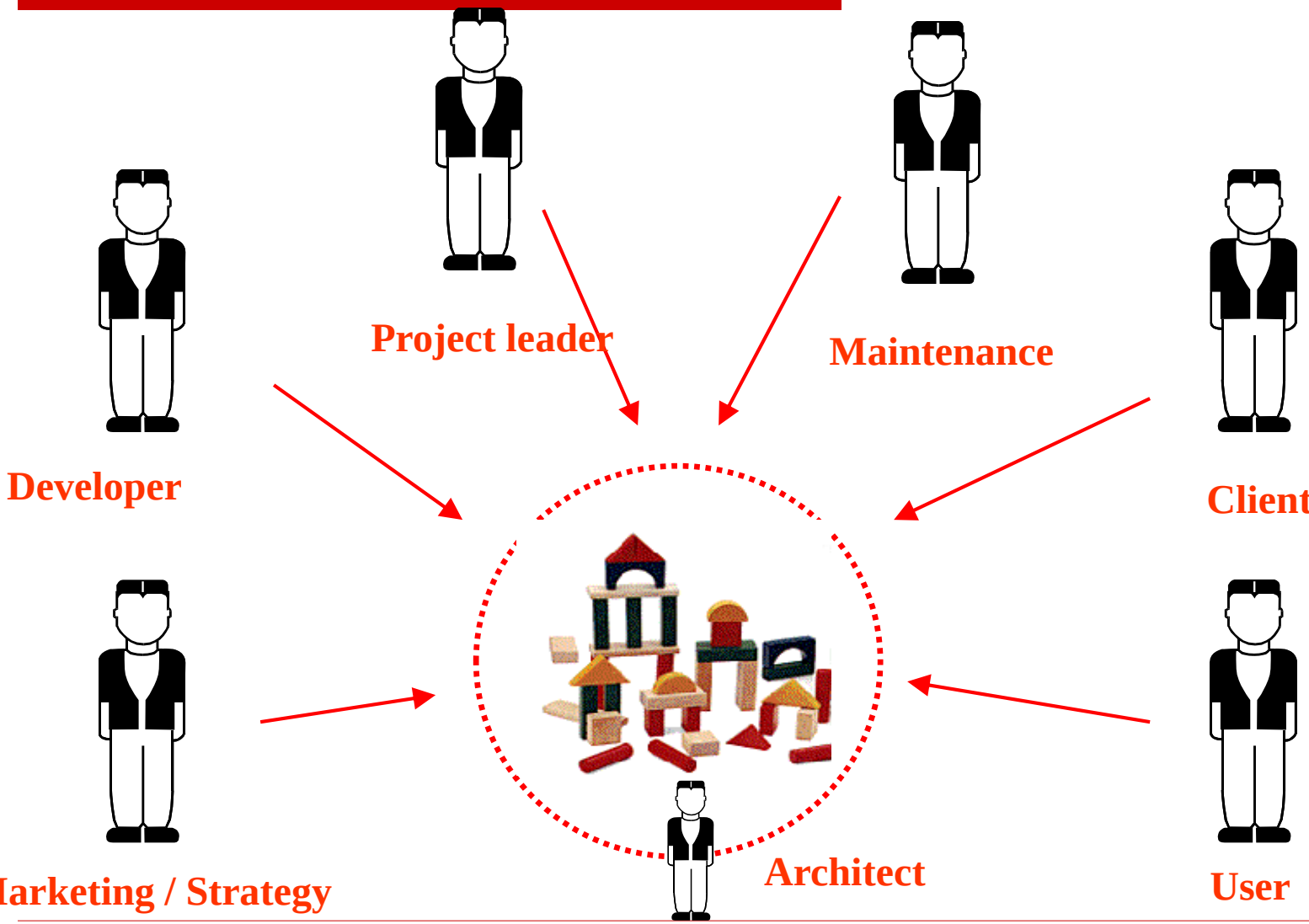
- Definition
- An important activity
- A difficult activity
- Conclusion

Multiple influences

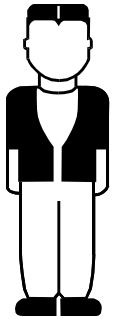


- Starting from the same requirements, two architects will end up with different architectures. Why ?

A meeting place !



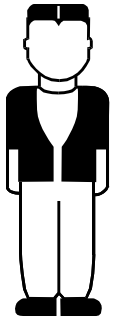
Diverging interests



Marketing / Strategy

- **Market tendency**
- **Coherence with company plans**
- **Coherence with company image (innovative, safe, …)**
- **Partnership coherence**

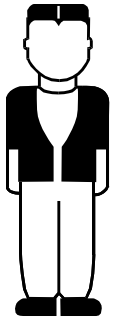
Diverging interests



Project leader

- **Overall simplicity**
- **Opportunity to assess progress**
- **Requirements meeting**
- **Mastered technology**
- **Structuration in line with existing teams**
- **Reuse of internal components and expertise**
- **Risk management**

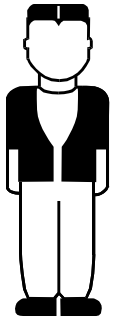
Diverging interests



Developer

- **Well defined components, simple interfaces**
- **Hot technology**
- **No constraints on their components**

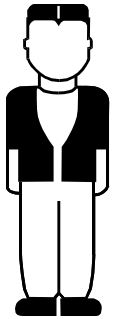
Diverging interests



Maintenance

- **Component isolation**
- **Administration oriented interfaces**
- **Known technologies**

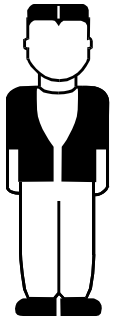
Diverging interests



Client

- **Guarantee of good behavior**
- **Efficient, cheap technologies**
- **Deadline meetings,**

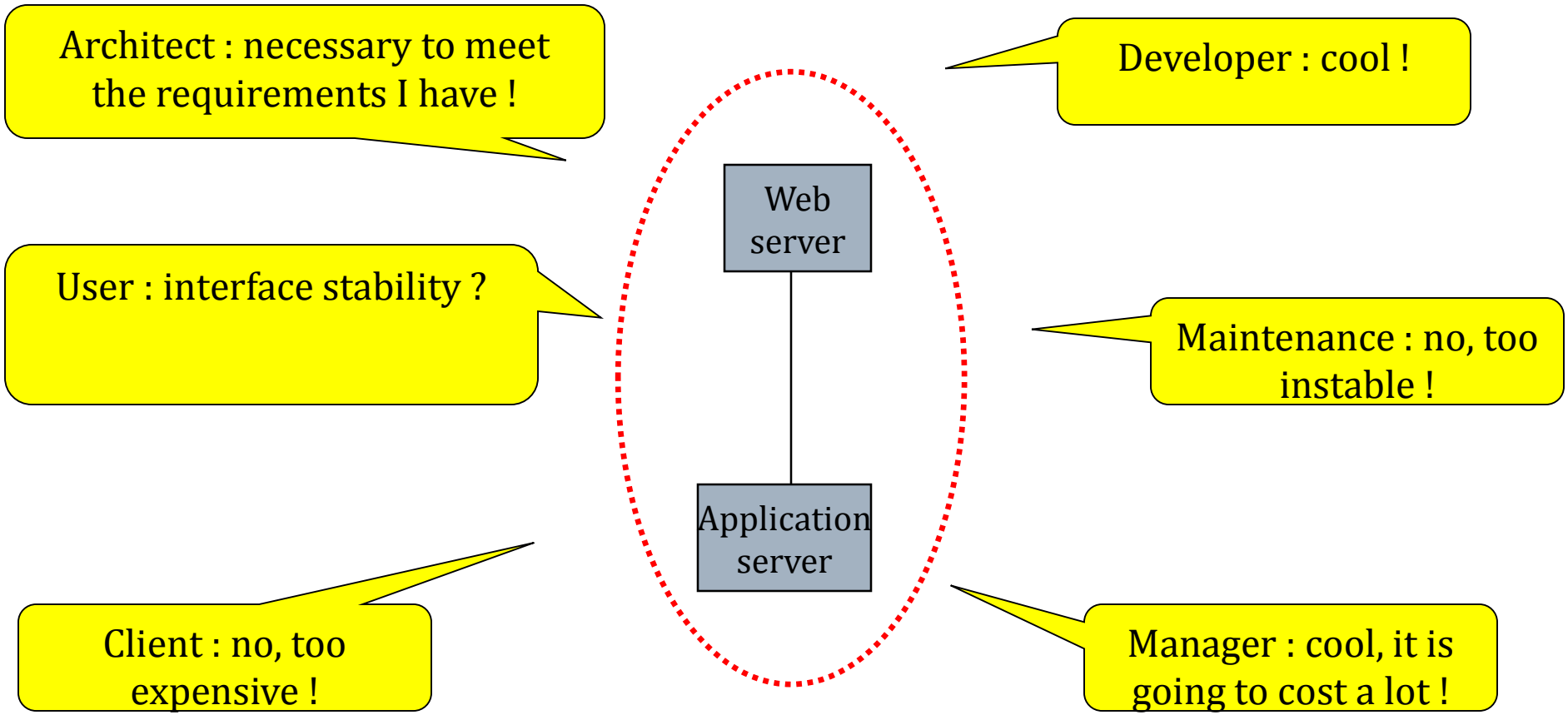
Diverging interests



User

- **Easiness of use**
- **Performance, security, robustness**

Example


Architect : necessary to meet the requirements I have !
Developer : cool !
Web server
User : interface stability ?
Maintenance : no, too instable !
Application server
Client : no, too expensive !
Manager : cool, it is going to cost a lot !

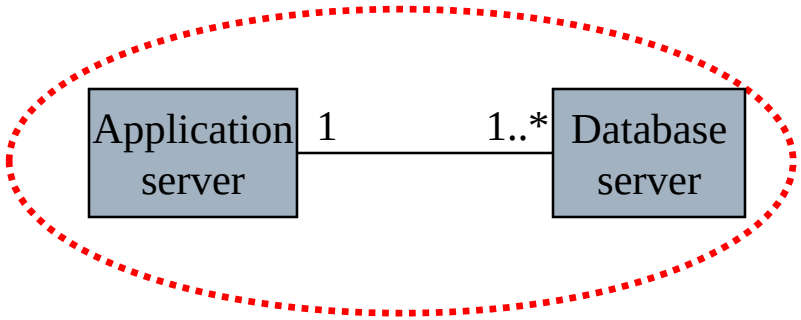
Organizational factors

- Architectural decisions are deeply influenced by the organization goals and strategy
  - Use of existing expertise and targeted expertise
  - Existing internal methods
  - Needs regarding reuse (to amortize past investments)
  - Partnerships

Example

ORACLE : DBMS rules. No need for an application server !

BULL : Application server matters. DBMS is not an issue.!



Microsoft : Ok, but forget about Java !

IBM : Good ! Both are needed (WS et DB2) !

Technological factors

- Technological current tendencies
  - DBMS
  - Web browser
  - Web services for integration
  - XML
  - … that was not the case some time ago
- There is a fashion factor !

Architect influence

- Architects use their past experiences
  - Good experiences will lead to reuse successful solutions
  - Bad experiences will lead them to give up related solutions
  - … but conditions have changed
- Tools, runtimes evolve a lot !

Outline

- Definition
- An important activity
- A difficult activity
- Conclusion

Major points

- An abstract specification
  - Based on components and connectors
- Conceptual *vs.* technical detail
  - Different levels of specification
- An architecture is a model
  - Hard to remain in sync with developments

Software architect

- The architect's role is to define architectures and to make them resilient
  - Project and organization level

- A hard job
  - High technical skills
  - Good knowledge of SE principles
  - High communication skills

Architecture importance

- Today recognized as a major step in software production
  - This is reinforced by outsourcing
- Creation of the software architect qualification
  - In many companies since 2000's
- Bill Gates was Microsoft chief architect!

Problems

- A recent, immature domain
  - No standard representation
  - No standard process (DAF, DAL, DAT, …)
  - No evaluation tools

- Many companies are struggling to define/install guiding processes

References

- Software architecture in practice - second edition
  Len Bass, Paul Clements, Rick Kazman
  Addison Wesley, 2003

- Pattern-oriented software architecture
  Buschmann, Meunier, Rohnert, Sommerlad, Stal
  Wiley, 1996

- Applied software architecture
  Hofmeister, Nord, Soni
  Addison Wesley, 2000

- Design and use of software architectures
  Jan Bosch
  Addison Wesley, 2000