Architecture Distribuée

Cours n°7

A.Saval

Objectifs du cours “Architectures distribuées”

- Compréhension des motivations
- Compréhension de la logique de conception d'une architecture distribuée
- Maîtrise des principaux modèles
- Aperçu des problèmes posés
- Aperçu de quelques frameworks existants

Aperçu du cours

- Introduction
- Problème de conception d'architecture
- Architecture logique & matérielle
- Système distribué
- Modèles d'architecture
  - Client/serveur
  - 3-tiers
  - N-tiers
  - Virtualisation

INTEGRATION

Intégration

Aggrégation d'un ensemble de sous-systèmes

- Verticale: création de silos indépendants
- Etoile: les sous-systèmes sont interconnectés
- Horizontale: bus dédié à la communcation

Intégration

- Difficile
- Source de conflits
- Plusieurs bonnes solutions
- Choix validés a posteriori
- Peu d'exemples réutilisables
- Méthodologies liées

Intégration

Système de messagerie asynchrone

- Couplage faible
- Latence
- Qualité
- Modèle de donnée commun

Réutilisation ?

Intégration par Pattern

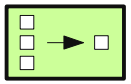
Détecter les modèles de réponses adaptés aux problèmes

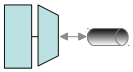
- Répond à un design particulier
- Rarement intuitif
- Compromis
- Expérimenté
- Réutilisable

Enterprise Integration Patterns

- Styles d'Intégration
- Systèmes de messageries
- Construction de Message
- Routage de Message
- Transformation de Message
- Endpoints de messagerie
- Gestion du Système

Enterprise Integration Patterns

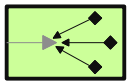

Aggregator

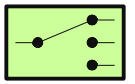

Channel Adapter

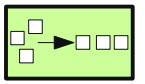

Channel

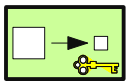

Channel Purger

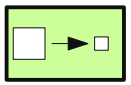
Command Message


Competing Consumers

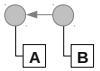

Content Based


Resequencer

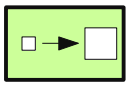

Claim Check

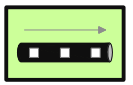

Content Filter


Control Bus

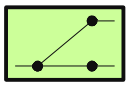


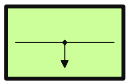
Correlation ID

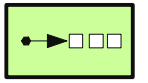

Content Enricher

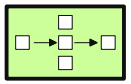

Datatype Channel


Dead Letter Channel

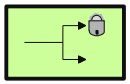

Detour

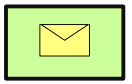

Wire Tap

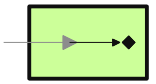

Test Message


Composed Message

Document Message

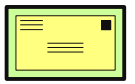

Durable Subscriber

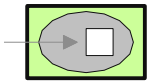

Envelope Wrapper

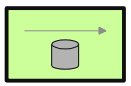

Event-Driven Consumer

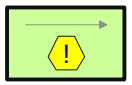
Event Message


Message Filter

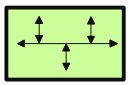

Return Address

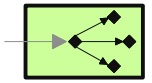

Transactional Client

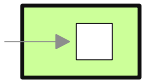

Guaranteed Delivery



Invalid Message


Message Bus

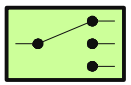

Message Dispatcher

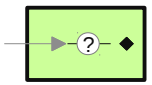

Message Endpoint


Message


Message Branch

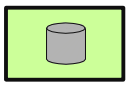

Blank

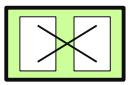

Message Router



Selective Consumer

Message Sequence


Message Store


Message Translator

Service Activator

Messaging Bridge

Routing Slip

Messaging Gateway

Normalizer

Poling Consumer

Process Manager

Publish-Subscribe Channel

Recipient List

Request Reply

Splitter

http://www.enterpriseintegrationpatterns.com/toc.html

Enterprise Integration Patterns

- Spring Integration
- Apache Camel
- Mule ESB
- Guaraná DSL
- Fabric8

Micro Service

- Totalement indépendant (Interface, service, protocol, stockage)
- Réplicable
- Déployable
- Support par une équipe dédiée
- Facilement Intégrable

Conclusion

- Chaque architecture répond à ses avantages et ses inconvénients.
- Ces différentes architectures peuvent être couplées

  Exemple: Modèle 3-tiers dont la persistence des données est gérée en Peer-to-Peer.
- Evolution des activités et des besoins =>

  Apparitions de nouvelles architectures

  (Recherches sur le flocking de données et la sémantique dans les graphs...)

Références

- Software Architecture: IEEE Standard 1471-2000
- P. Kruchten, Architectural Blueprints—The “4+1” View Model of Software Architecture, IEEE Software 12 (6), Nov. 1995, pp42-50
- Tanenbaum & van Steen, Distributed Systems, Principles and Paradigms, seconde édition
- Architecture of Distributed Systems, cours de Johan Lukkien, 2011
- Architectural Patterns Revisited – A Pattern Language, Paris Avgeriou & Uwe Zdun, 2005
- Software Architecture, Foundations, Theory, and Practice, R.N. Taylor, N. Medvidovic, E.M. Dashofy, Wiley & Sons, 2009
- Software Architecture in Practice, Second Edition, L. Bass, P. Clements, R. Kazman, SEI Series in Software Engineering, Addison-Wesley, 2003