

DÉVELOPPEMENT D'APPLICATIONS RÉPARTIES

©Amen Ben Hadj Ali

amenbha@hotmail.com

Chapitre 0 - Présentation du cours

2

○ Objectifs du cours

- Etudier les caractéristiques des systèmes répartis et montrer leurs différences avec les systèmes centralisés.
- Principes de base des systèmes répartis.
- Apprendre aux étudiants les techniques de résolution des problèmes liés à la répartition.
- Apprendre aux étudiants les méthodes et techniques pour la programmation et l'exploitation
- d'applications réparties.

Chapitre 0 - Présentation du cours

3

Pré requis

Programmation objet, méthodologies de conception, fichiers et BD, téléinformatique

Chapitre 1- Introduction aux Systèmes Répartis

4

Plan

1 Notions de base

2 Architecture répartie

L'informatique **centralisée** correspond à la structure hiérarchique d'une entreprise avec sa nécessité de contrôle et de coordination

Evolution de la technologie

- ↑ des performances
- ↓ des coûts
Rapport coût/performance des stations
- Banalisation des réseaux de télécommunication
- Performances des voies de télécom (débit et fiabilité)

Evolution des besoins

- Structure des entreprises et des organisations : communication et partage
- Accès universel à l'information

Applications réparties

1 Définitions

6

Pas de consensus...

Définition 1 [Tanenbaum] : Système Réparti

Un ensemble d'ordinateurs indépendants qui apparaît à un utilisateur comme un système unique et cohérent

- Les machines sont autonomes
- Les utilisateurs ont l'impression d'utiliser un seul système (**la transparence**).

Définition 2 [Lamport]: Un système réparti est un système qui vous empêche de travailler quand une machine dont vous n'avez jamais entendu parler tombe en panne.

1 Définitions

7

Définition

Un Système Réparti (distribué) est un ensemble d'entités autonomes de calcul (ordinateurs, processeurs, processus, processus léger etc.) interconnectées et qui peuvent communiquer par un réseau de communication et qui communiquent par envoi de messages.

Exemples :

- réseau physique de machines
- Un logiciel avec plusieurs processus sur une même machine.

Réparti = distribué ?

Distribué → fabricants d'ordinateurs

Réparti → Génie Logiciel

1 Pourquoi les SR?

8

Objectifs généraux

- Aspects économiques (rapport prix/performance)
- Adaptation de la structure d'un système à celle des applications (géographique ou fonctionnelle)
- Besoin d'intégration (applications existantes)
- Besoin de communication et de partage d'information
- Réalisation de systèmes à haute disponibilité
- Partage de ressources (programmes, données, services)
- Réalisation de systèmes à grande capacité d'évolution

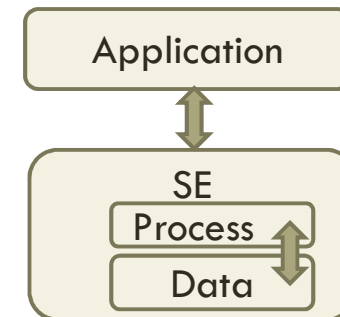
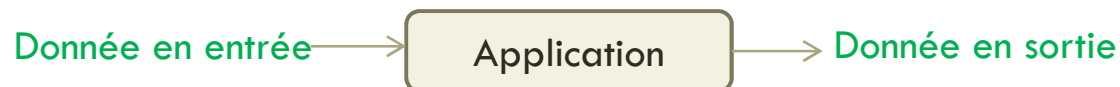
Objectifs spécifiques de l'Informatique Repartie

- Proposer des méthodes et outils pour simplifier le développement d'application réseau client/serveur, en essayant de s'abstraire de l'aspect "distant" : proposer une programmation "naturelle"
- Pour les applications "lourdes" :
 - ▣ Décomposer les applications en ensembles de services
 - ▣ Rationaliser la répartition des services pour limiter les échanges d'informations

1 Définitions

9

- une **application informatique** est un programme exécutable sur une machine ou plusieurs machines qui représente la logique de traitement des données manipulées par l'application.
- Elle s'exécute en mémoire vive au dessus d'un système d'exploitation
- Avant exécution, elle est stockée sur un support persistant (DD)
- L'application émet un résultat sous forme de données soit affichées, soit enregistrées sur un disque.



- Dans ce schéma, les traitements, les données d'entrées, les données de sortie sont sur une seule machine.
- Un grand nombre d'applications ne s'exécutent pas intégralement sur un seul noeud de calcul Il s'agit d'applications réparties (distribuées)
- Application **répartie** = traitements **coopérants** sur des données **réparties**.

1 Système vs application

10

- **Système** : gestion des ressources communes et de l'infrastructure, lié de manière étroite au matériel sous-jacent
 - ▣ Système d'exploitation : gestion de chaque élément
 - ▣ Système de communication : échange d'information entre les éléments
 - ▣ Caractéristiques communes : cachent la complexité du matériel et des communications, fournissent des services communs de plus haut niveau d'abstraction
- **Application** : réponse à un problème spécifique, fourniture de **services** à ses utilisateurs (qui peuvent être d'autres applications). Utilise les services généraux fournis par le système

1 Services et interfaces

11

Définition

- Un système est un ensemble de composants (au sens non technique du terme) qui interagissent
- Un service est “un comportement **défini** par **contrat**, qui peut être implémenté et fourni par un composant pour être utilisé par un autre composant, sur la base exclusive du contrat”

Mise en oeuvre

- Un service est **accessible** via une ou plusieurs **interfaces**
- Une interface décrit l'interaction entre client et fournisseur du service
 - ▣ Point de vue opérationnel : définition des opérations et structures de données qui concourent à la réalisation du service
 - ▣ Point de vue contractuel : définition du contrat entre client et fournisseur

1 Interface (1/2)

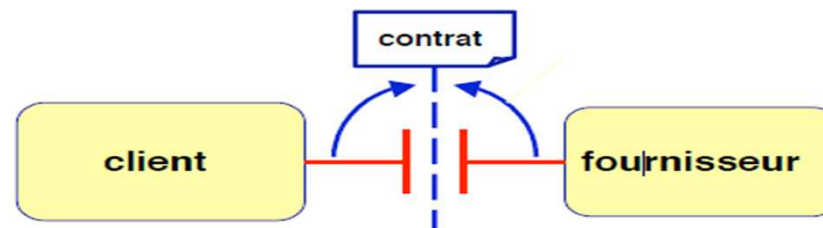
12

La fourniture d'un service met en jeu deux interfaces :

- Interface **requise** (côté client)
- Interface **fournie** (côté fournisseur)

Le contrat spécifie la compatibilité (conformité) entre ces interfaces

- Au delà de l'interface, chaque partie est une “boîte noire” pour l'autre (principe d'encapsulation)
- Conséquence : client ou fournisseur peuvent être remplacés du moment que le composant remplaçant respecte le contrat (est conforme)



1 Interfaces(2/2)

13

- Partie “opérationnelle”

Interface Definition Language (IDL)

Pas de standard, mais s’appuie sur un langage existant

- IDL CORBA sur C++
- Java et C# définissent leur propre IDL

- Partie “contractuelle”

Plusieurs niveaux de contrats

- ▣ Sur la forme : spécification de types → conformité syntaxique
- ▣ Sur le comportement (1 méthode) : assertions → conformité sémantique
- ▣ Sur les interactions entre méthodes : synchronisation
- ▣ Sur les aspects non fonctionnels (performances, etc.) : contrats de QoS

1 Services répartis

14

- Services de désignation
 - ▣ Nommage des objets
 - ▣ Localisation des objets à travers le SR
- Services de communication
 - ▣ Exécution de requetes C/S
 - ▣ Exécution de req RPC
- Services de fichiers répartis
 - ▣ Accès transparent au fichiers
 - ▣ Partage de fichiers
- Services d'exécution répartie
 - ▣ Gestion des accès concurrnets et des interblocages
- Services de transactions réparties
 - ▣ Respect des propriétés ACID
- Services de temps réparti
 - ▣ Construction et maintien d'un état global
- Services de mémoire répartie

1 Exemple

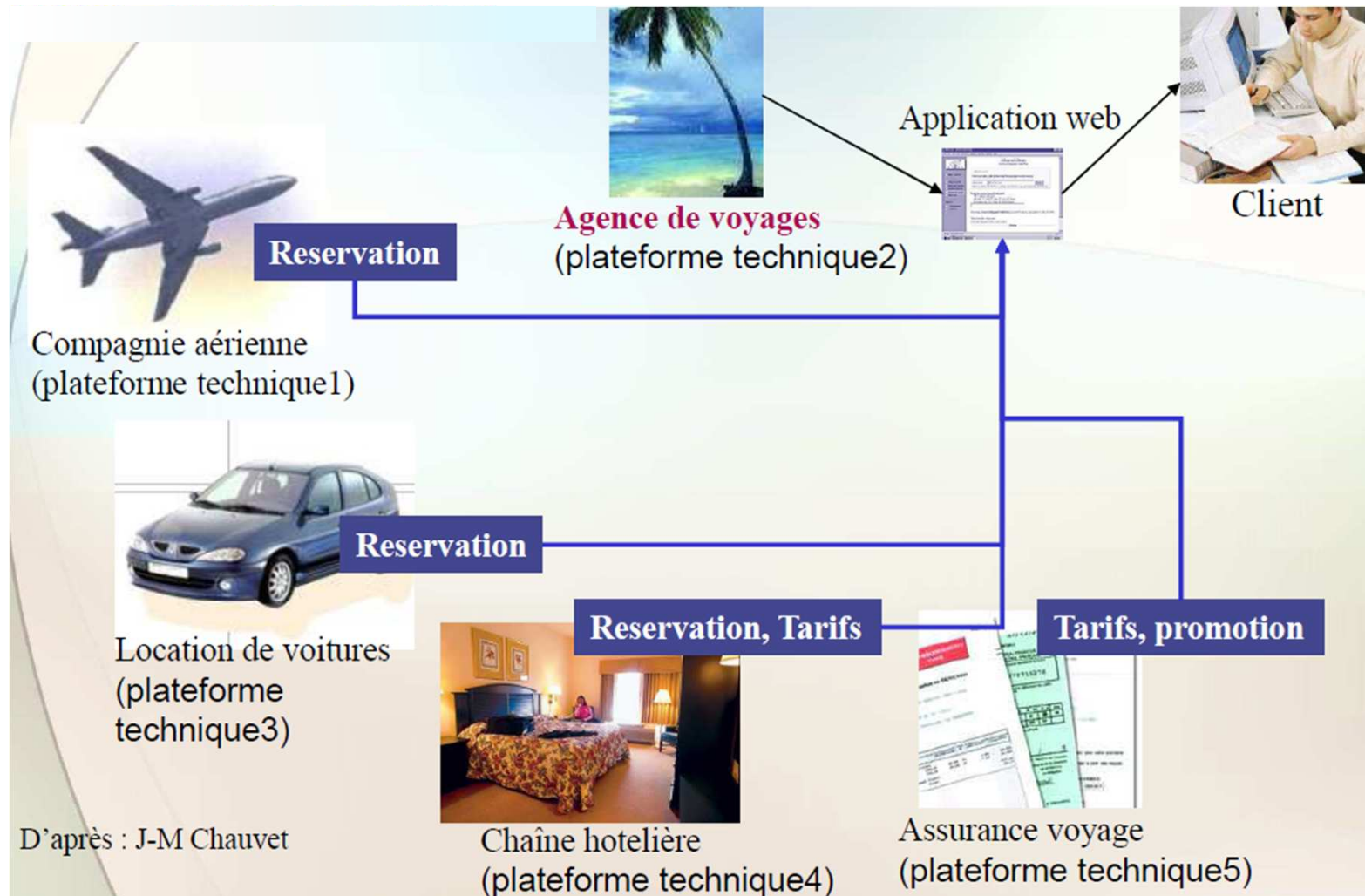
15

Agence de voyage

- Un produit « voyage » = la combinaison de plusieurs produits
 - Gestion de réservation de billets de transport
 - Gestion de réservation des hôtels
 - Gestion de réservation de voitures de location
 - ...
- le résultat d'informations récupérées auprès de différents fournisseurs :
 - Compagnies aériennes
 - Chaines hôtelières
 - Agences de location de voitures
 - ...

1 Exemple

16



1 Domaines d'application

17

- Coordination d'activités :
 - Systèmes à flots de données (workflow).
 - Systèmes à agents.
- Communication et partage informations :
 - Bibliothèques virtuelles.
- Applications temps réel :
 - Contrôle de procédés industriels.
 - Avionique, etc.
 - Localisation de mobiles.
- Toutes les applications qui nécessitent des utilisateurs ou des données réparties.

1 Informatique répartie vs informatique centralisée

18

Informatique centralisée	Informatique répartie
Un seul OS	Différents OS
Espace mémoire partagé	Mémoires individuelles
Pas de réseau	Pbs de communication: réseau, firewall...
Service local	Retrouver le service? Qui le propose? Où se trouve t il?
Un seul langage de programmation	Différents langages: représentations différentes de l'info., passage de paramètres, comment gérer les erreurs, différents paradigmes
Les users travaillent sur une seule machine	Les users ont l' <i>impression</i> de travailler sur une seule machine : la transparence

Que doit résoudre un système réparti?

- Pas d'état global: Non connaissance de l'état global instantané du système
- Pas d'horloge globale
- Fiabilité et sécurité relative

Chapitre 1- Introduction aux Systèmes Répartis

19

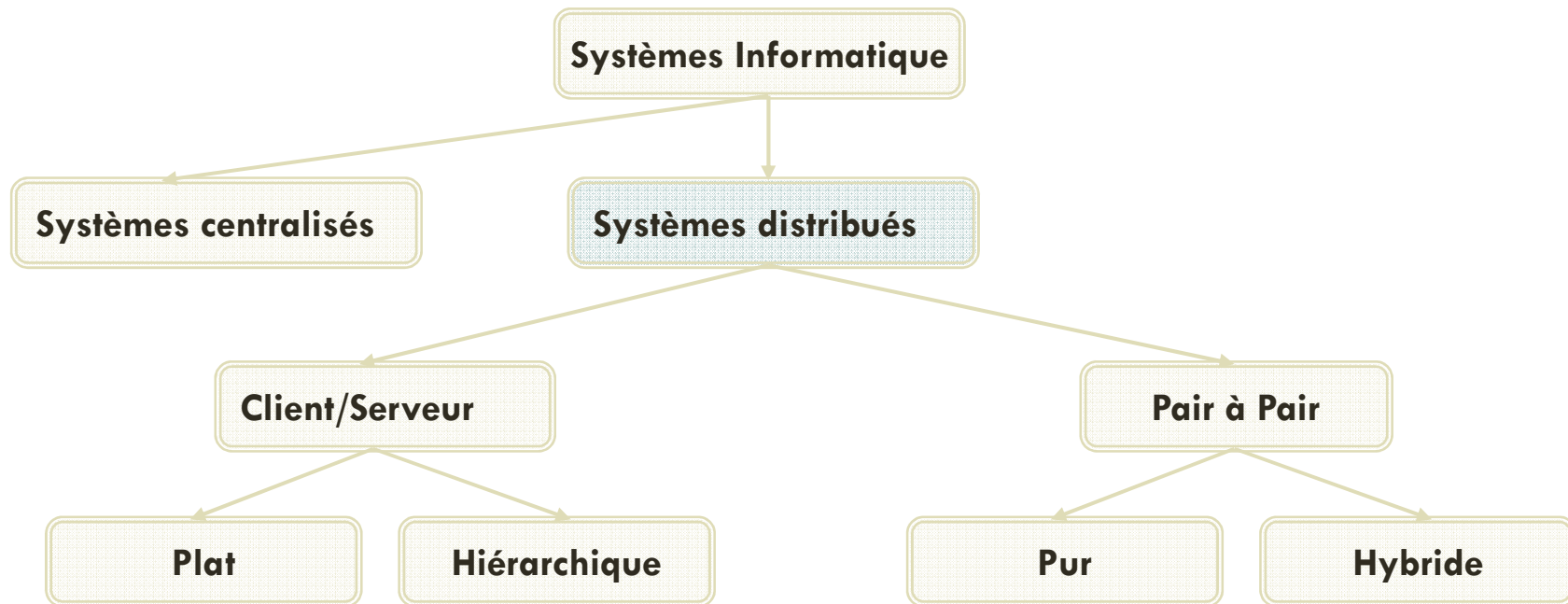
Plan

1

Notions de base

2

Architecture répartie



2 Architecture répartie

21

Rappel : Système réparti : ensemble de composants (matériel ou logiciel)

- ▣ interconnectés par un réseau de communication,
- ▣ qui coopèrent pour l'exécution de tâches communes,
- ▣ dont l'organisation fait montre de robustesse.
- Propriétés demandées aux systèmes répartis (rappel)
- Transparence
 - ▣ utilisation de service distant comme local
 - ▣ indépendance à la localisation physique
 - ▣ mobilité des informations et services
- Abstraction (séparation interface/réalisation)
- Récupération de l'existant
- Capacité d'évolution

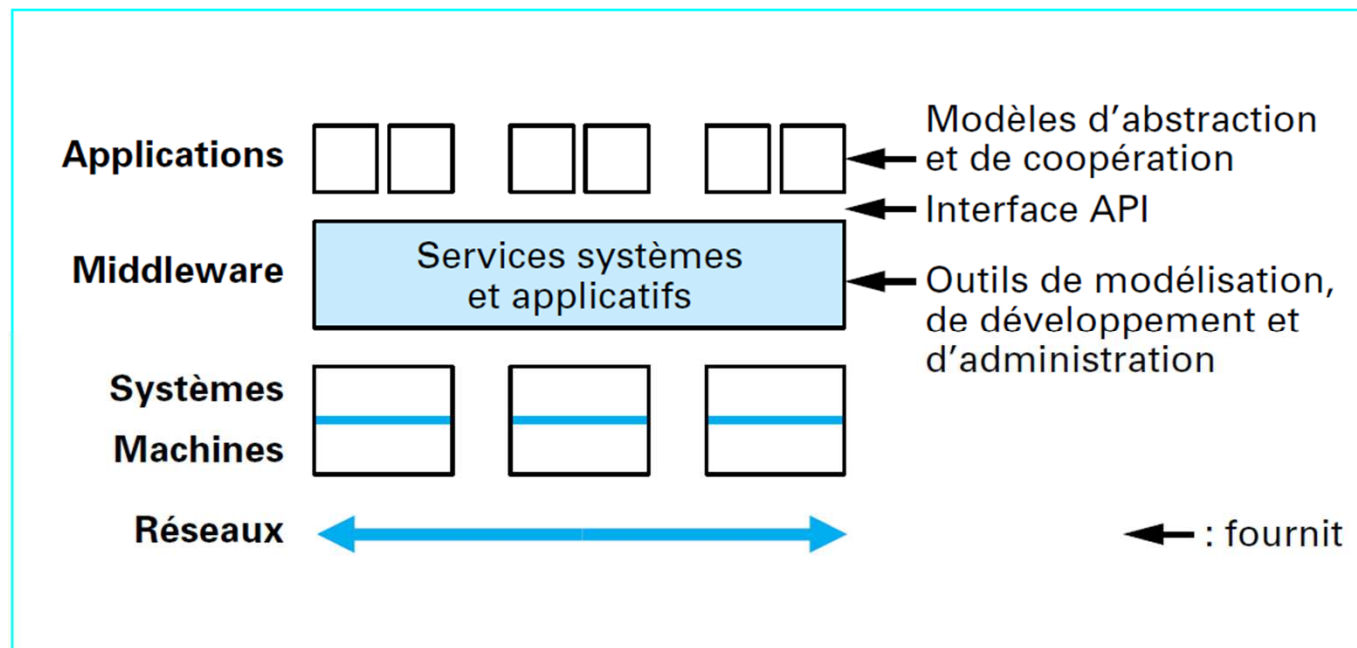
- Organisation
- Coordination
- Communication
- Sécurité
- Disponibilité
- Extensibilité : capacité de croissance
- Génie logiciel : construction + administration

- Client-serveur
 - ▣ 1er modèle. Schéma de base synchrone.
 - ▣ Technologies adaptées : RPC, Corba.
- Objets partagés (organisations diverses)
 - ▣ Technologies adaptées : Corba, RMI.
- Flots de communication
 - ▣ Discrets (messages) ou continus (multimédia)
- Code mobile
 - ▣ transporter avec les données les programmes de traitement
 - ▣ Problèmes de sécurité
- Agents
 - ▣ Programme agissant pour le compte d'une entité cliente
 - ▣ Agents fixes ou mobiles, statiques ou adaptatifs
 - ▣ Coopération entre agents

2 Organisation: une structure commune

24

- Le **middleware** est une couche intermédiaire entre le système d'exploitation et l'application afin d'en améliorer sa mise en oeuvre : L'objectif principal du middleware est d'unifier, pour les applications, l'accès et la manipulation de l'ensemble des services disponibles sur le réseau, afin de rendre l'utilisation de ces derniers presque **transparente**.



- Le **middleware = intergiciel** est une classe de logiciels systèmes agissant en qualité d'infrastructure pour le développement et le déploiement d'applications réparties : exp. CORBA
- Le développement du middleware impose une normalisation des interfaces
 - Logiciel de base
 - Domaine spécifiques d'applications
- **Nombreux standards et consortiums**
 - Open Group (ex OSF) : systèmes, outils de base
 - Web Consortium (W3C) : Web + outils associés
 - OMG : objets répartis (CORBA, IIOP)+ Modèles (MDA)

2 Types de middleware

26

- **Les middleware d'accès aux données:**
 - SQL*Net : Interface propriétaire permettant de faire dialoguer une application cliente avec une base de données Oracle
 - ODBC, JDBC, ADO : Interface standardisée isolant le client du serveur de données
- **Les middleware à Objets répartis CORBA, JAVA RMI (Remote Method Invocation)**
 - Principe : Appel d'une procédure qui s'exécute sur un site distant
 - invocation et attente de réponse (appel synchrone)
- **Le middleware à composants (EJB, COM/DCOM, web services)**
- **Les middlewares transactionnels (Tuxedo): Coordonner les transactions distribuées**
- **MOM (message oriented middleware)**
 - File d'attente de message (ex : MQSeries d'IBM)

2 Coordination

27

- Permettre aux composants d'une application d'interagir en respectant des règles fixées
 - ▣ Englobe synchronisation et communication
 - ▣ Divers niveaux d'abstraction
 - ▣ Divers mécanismes (messages, événements, etc...)
- Caractéristiques de la communication
 - ▣ Synchrone : schéma prédéfini (RPC)
 - ▣ Asynchrone : réaction à un événement
 - ▣ Flots discrets ou continus – QoS
- Informations partagées
 - ▣ Espace unique ou multiples
 - ▣ Divers modes de réalisation (centralisé, réparti, dupliqué,...)

2 Sécurité

28

- Confidentialité
- Intégrité
 - ▣ Protection par contrôle des droits d'accès
 - ▣ Isolation (pare-feux)
- Authentification, signature électronique
 - ▣ Identification des partenaires
 - ▣ Non-déni d'envoi ou de réception
 - ▣ Messages authentifiés
 - ▣ Respect possible de l'anonymat
- Technique de base : cryptographie
 - ▣ Aspects techniques
 - ▣ Aspects sociaux, juridiques et législatifs

2 Disponibilité

29

- Des informations/ressources/services
- Séparation des machines « contrôlables » (serveurs) et les « non contrôlables » (les autres)
- Technique de base n°1 : la duplication
 - ▣ Des serveurs
 - ▣ Des services
 - ▣ Des informations
- Technique de base n°2 : les protocoles de groupe
 - ▣ Protocole d'appartenance
 - ▣ Protocole de diffusion (fiable, atomique, causale)
- Technique de base n°3 : les transactions

2 Extensibilité

30

- Propriété souhaitable (scalability): éviter la dégradation des performances lorsqu'augmentent:
 - ▣ Le nombre de sites
 - ▣ Le nombre d'utilisateurs
 - ▣ Le volume de données
 - ▣ La fréquence des interactions
- Quelques heuristiques
 - ▣ Éviter la concentration des données
 - ▣ Éviter les algorithmes nécessitant la participation de tous les sites
 - ▣ Déléguer et décentraliser les décisions
- Un exemple réussi : le DNS (Domain Name System)

- Réutilisation de l'existant
- Adaptation à l'environnement
 - ▣ Support matériel (machines et réseaux)
 - ▣ Hétérogénéité
 - ▣ Conditions variables d'utilisation (charge, ressources, ...)
- Quelques voies
 - ▣ Méthodes à base d'objets (« composants »)
 - ▣ Encapsulation, interfaces séparées
 - ▣ Spécialisation de classes
 - ▣ Interfaces d'administration
 - ▣ ...