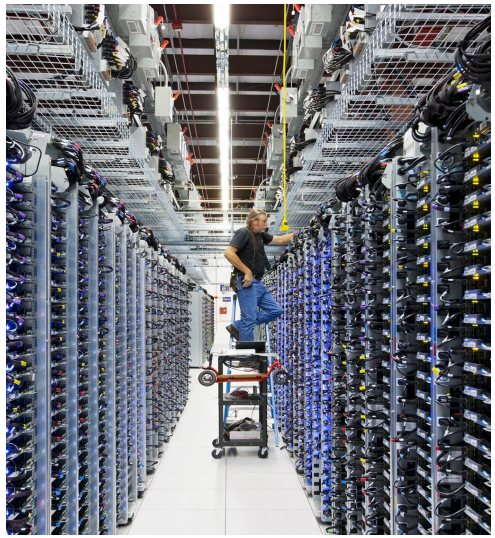
heig-vd

HEIG-VD | TIC – Technologies de l'Information et de la Communication

Administration IT — Systèmes à haute disponibilité



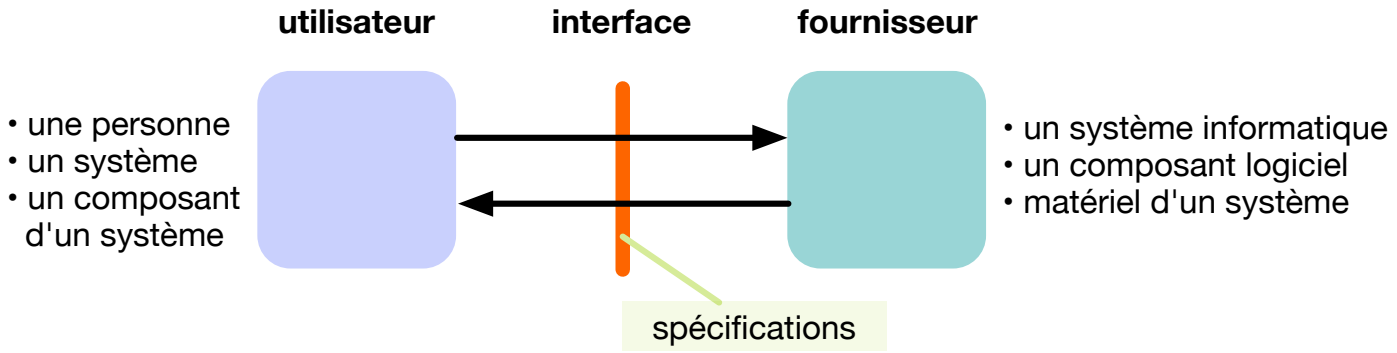



Année académique 2015/16

Tolérance aux fautes

Définitions de base

- **Service** : Ensemble de fonctions défini par une **interface**, « contrat » entre le fournisseur et l'utilisateur du service.

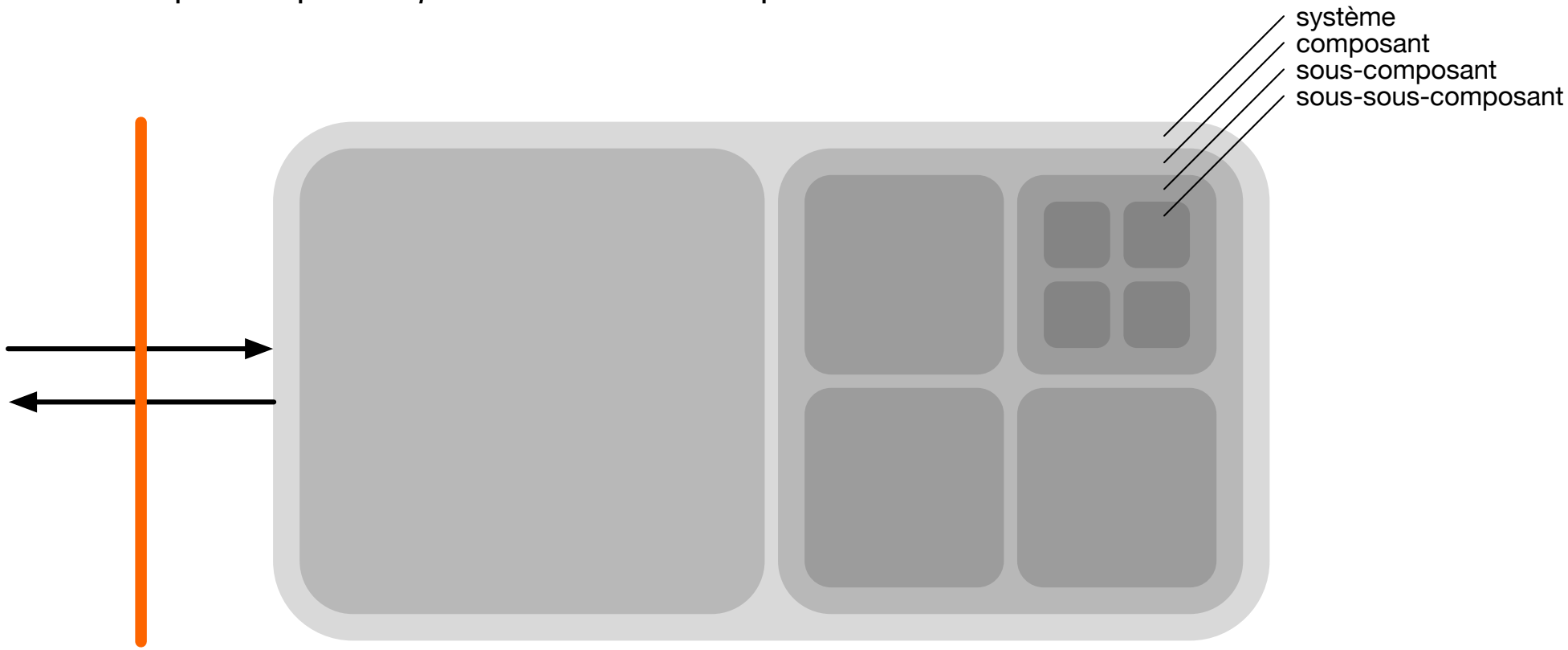


- Propriétés attendues d'un service
  - Propriétés fonctionnelles (définies dans la spécification d'interface)
    - Validité *(correctness)* : Le système est conforme à ses spécifications. Propriétés de sûreté et vivacité.
  - Propriétés non-fonctionnelles (les autres)
    - Performances
    - Sûreté de fonctionnement
  - La distinction n'est pas toujours évidente

Tolérance aux fautes

Système et composants

- Un système peut être composé de composants qui fournissent des *services* au système. Pour fournir les services, le composant peut nécessiter des services d'autres composants
  → un composant peut *dépendre* d'autres composants.



Tolérance aux fautes

Sûreté de fonctionnement

- La **sûreté de fonctionnement** *(dependability)* d'un système est la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service qu'il leur délivre.
  Elle est perçue à travers différents attributs :
  - **Disponibilité** (availability) : Mesure le fait que le service (correct) soit *prêt* pour l'utilisateur
    - Mesure : Fraction du temps (sur une période déterminée) durant laquelle le système fournit le service.
  - **Fiabilité** (reliability) : Mesure la *continuité* du service (correct)
    - Mesure : Probabilité (fonction du temps t) que le système ne soit pas défaillant entre le temps 0 et le temps t.
  - **Sécurité-innocuité** (safety) : Mesure la non-occurrence de conséquences catastrophiques des défaillances sur l'environnement
    - Il faut définir « catastrophique » et « environnement »
  - **Maintenabilité** (maintainability) : Mesure l'aptitude à *réparer* (et faire évoluer) les systèmes
  - **Sécurite** (security) : Concerne la confidentialité et l'intégrité de l'information.

Tolérance aux fautes

Sûreté de fonctionnement

- **Il n'y a pas de critère absolu**
  - L'importance relative des critères dépend
    - de la nature de l'application
    - des exigences des utilisateurs
    - des conditions d'utilisation (environnement, etc.)
- Exemples
  - Système embarqué : fiabilité, disponibilité
  - Système de communication (ex. commutateur téléphonique) : disponibilité
  - Service de fichiers, base de donnée : disponibilité, sécurité
  - Système de transport (ex. navigation, guidage, freinage) : sécurité-innocuité, disponibilité

Tolérance aux fautes

Impact de l'indisponibilité sur le rendement des entreprises

Source: Hennessy, Patterson - Computer Architecture

Application	Coût d'indisponibilité par heure	Pertes annuelles avec un temps d'indisponibilité de		
		1% (87.6 h/année)	0.5% (43.8 h/année)	0.1% (8.8 h/année)
Brokerage operation	$6'450'000	$565'000'000	$283'000'000	$56'500'000
Credit card authorization	$2'600'000	$228'000'000	$114'000'000	$22'800'000
Package shipping services	$150'000	$13'000'000	$6'600'000	$1'300'000
Home shopping channel	$113'000	$9'900'000	$4'900'000	$1'000'000
Catalog sales center	$90'000	$7'900'000	$3'900'000	$800'000
Airline reservation center	$89'000	$7'900'000	$3'900'000	$800'000
Cellular service activation	$41'000	$3'600'000	$1'800'000	$400'000
Online network fees	$25'000	$2'200'000	$1'100'000	$200'000
ATM service fees	$14'000	$1'200'000	$600'000	$100'000

Tolérance aux fautes

Terminologie

Terme	Description
Défaillance (failure)	Un état dans lequel le service délivré n'est pas conforme au cahier des charges / la spécification
Erreur (error)	Partie d'un composant qui peut conduire à une défaillance
Faute (fault)	La cause d'une erreur

Faute → Erreur → Défaillance

Tolérance aux fautes

Terminologie — Exemple

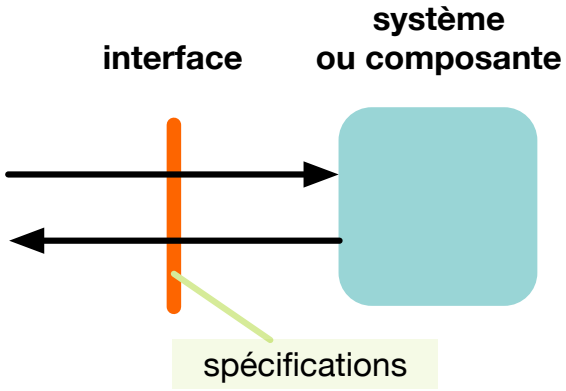
- Donner un exemple de défaillance. Identifier l'erreur cause de cette défaillance et la faute cause de l'erreur.
  - Système : le code dans un noeud (commutateur, routeur, ...) d'un réseau de télécommunications. Dans ce code, le programmeur a écrit « i = 0; » au lieu de « i = 1; », qui était l'instruction correcte. On a une faute.
  - À un moment particulier, l'ordinateur exécute cette instruction, et suite à un certain calcul, un tampon est dimensionné à 10 au lieu d'être dimensionné à 100. On a une erreur.
  - Les conditions d'utilisation du réseau font qu'exceptionnellement, ce jour-là, la charge est telle que le tampon reçoit un trafic trop important. Il y a alors trop de pertes (plus que les niveaux maximaux spécifiés). On a une défaillance.
- Responsable ?

Tolérance aux fautes

Défaillances

- Définition défaillance : Un système (ou composant) est sujet à une défaillance *(failure)* lorsque son comportement n'est pas conforme à sa spécification
  - Synonyme de défaillance : panne

- Remarques :
  - Le système ou composant est considéré comme « boîte noire » ; on ne regarde que son comportement global, observé à l'interface.
  - On peut définir différents degrés de gravité de défaillance en fonction de leur impact sur la sûreté de fonctionnement



Tolérance aux fautes

Classification des défaillances

- Panne franche (dit aussi arrêt sur défaillance, *fail stop*)
  - Seulement deux possibilités :
    - ou bien le système fonctionne et donne un résultat correct
    - ou bien il est en panne (défaillant), et ne fait rien
  - C'est le cas le plus simple, et on essaie de s'y ramener (au besoin en forçant l'arrêt d'un composant dès qu'une erreur y a été détectée : technique *fail fast*)
- Panne par omission
  - Le système perd des messages entrants (omission en réception), sortants (omission en émission), ou les deux. Il n'y a pas d'autres déviations par rapport aux spécifications.
  - Ce modèle peut servir à représenter des défaillances du réseau.
  - Plus difficile à traiter que la panne franche.

Tolérance aux fautes

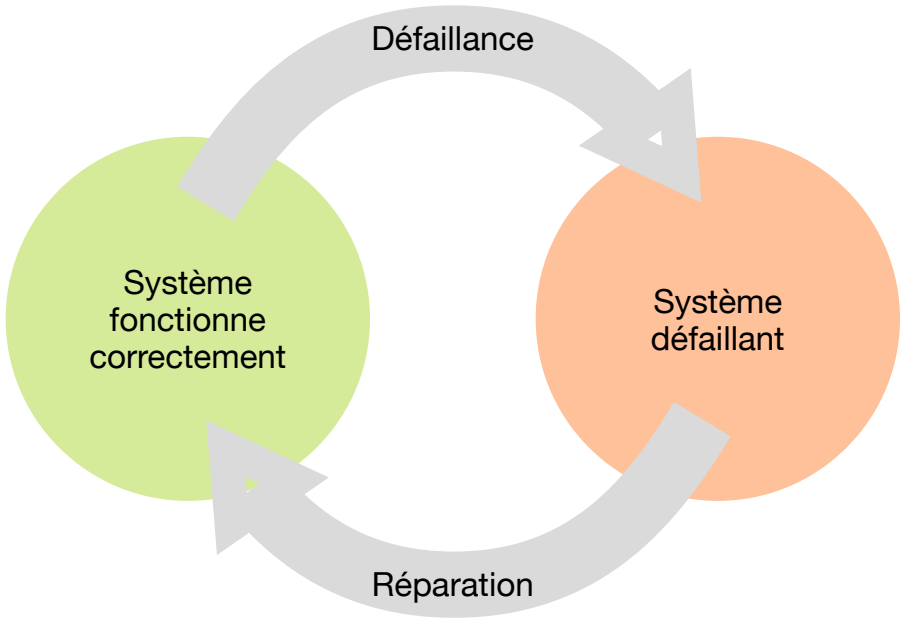
Classification des défaillances (suite)

- Pannes de temporisation
  - Les déviations par rapport aux spécifications concernent uniquement le temps.
    - Par exemple temps de réaction à un événement
- Pannes arbitraires (ou byzantines)
  - Le système peut faire « n'importe quoi » (y compris avoir un comportement malveillant)
  - Intérêt théorique : conditions les plus défavorables
  - Hypothèse parfois nécessaire pour des systèmes à très haute fiabilité dans un environnement hostile (nucléaire, spatial)
  - Traitable, mais nécessite une redondance élevée (typiquement $3k+1$ exemplaires d'un composant pour résister à k défaillances)

Tolérance aux fautes

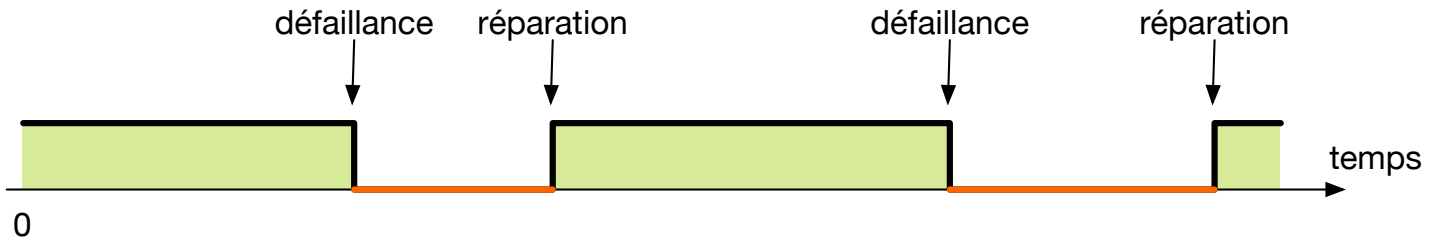
Mesures de fiabilité et disponibilité

- Pour pouvoir définir des mesures de fiabilité et disponibilité on suppose que le système (ou le composant) peut se trouver dans seulement deux états :
  - Le système fonctionne correctement (selon sa spécification)
  - Le système est défaillant
- La transition du fonctionnement correct au système défaillant est dû à une **défaillance**.
- Une **réparation** ramène le système à l'état correct.



Tolérance aux fautes

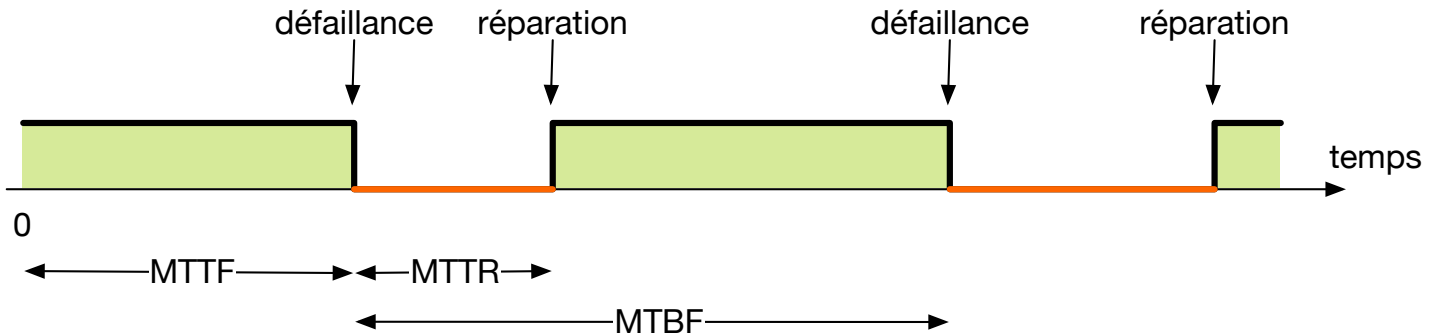
Mesures de fiabilité et disponibilité (suite)



- Pour simplifier les calculs on fait souvent les hypothèses suivantes :
  - Le système est observé depuis un instant initial de référence (temps 0).
  - Quand le système devient défaillant, il est réparé et devient comme neuf.
  - Le temps jusqu'à la défaillance est une variable aléatoire avec distribution exponentielle, c.à.d. la probabilité de défaillance ne dépend pas de l'âge.
  - Les défaillances de différents modules dans un système sont indépendantes.

Tolérance aux fautes

Mesures de fiabilité et disponibilité (suite)

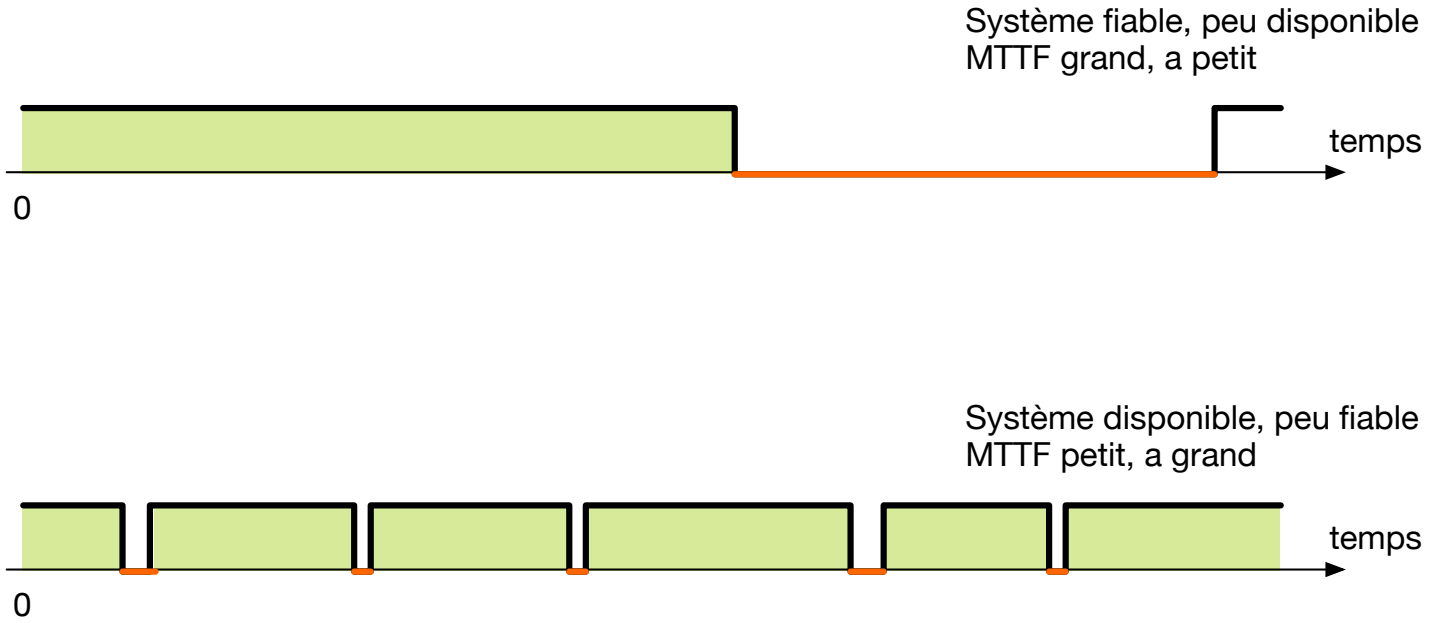


- Mesure de la fiabilité *(reliability)*
  - Probabilité *R(t)* que le système ne soit pas défaillant entre 0 et *t*
  - Temps moyen jusqu'à la prochaine panne *(Mean Time To Failure,* ***MTTF****)* : *MTTF = E(R(t))*
- Mesure de la disponibilité *(availability)*
  - Disponibilité instantanée : Probabilité *A(t)* que le système soit disponible (fournisse un service correct) à l'instant *t*
  - Disponibilité moyenne : Fraction moyenne du temps où le système est disponible (sur une période donnée) *a = E(A(t))*
- Mesure du temps de réparation
  - Temps moyen de réparation *(Mean Time To Repair, MTTR)*

Tolérance aux fautes

Différence entre fiabilité et disponibilité

- Bien comprendre la différence entre
  - fiabilité (mesurée par MTTF) et
  - disponibilité (mesurée par a = MTTF / (MTTF + MTTR))



Tolérance aux fautes

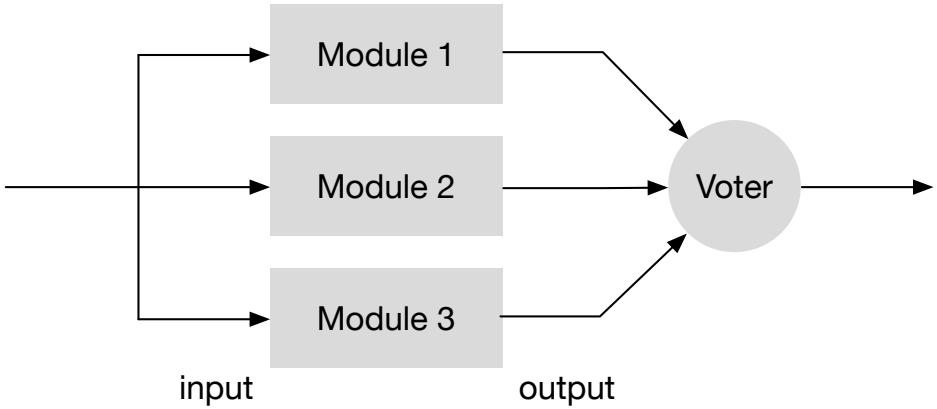
Disponibilité — "Nombre de neufs"

Non-disponibilité (par année)	Non-disponibilité	Disponibilité	Nombre de neufs
37 days	10%	90%	1
3.7 days	1%	99%	2
8.8 hours	0.1%	99.9%	3
53 min	0.01%	99.99%	4
5.3 min	0.001%	99.999%	5
32 sec	0.0001%	99.9999%	6
3.2 sec	0.000001%	99.99999%	7

Triple Modular Redundancy (TMR)

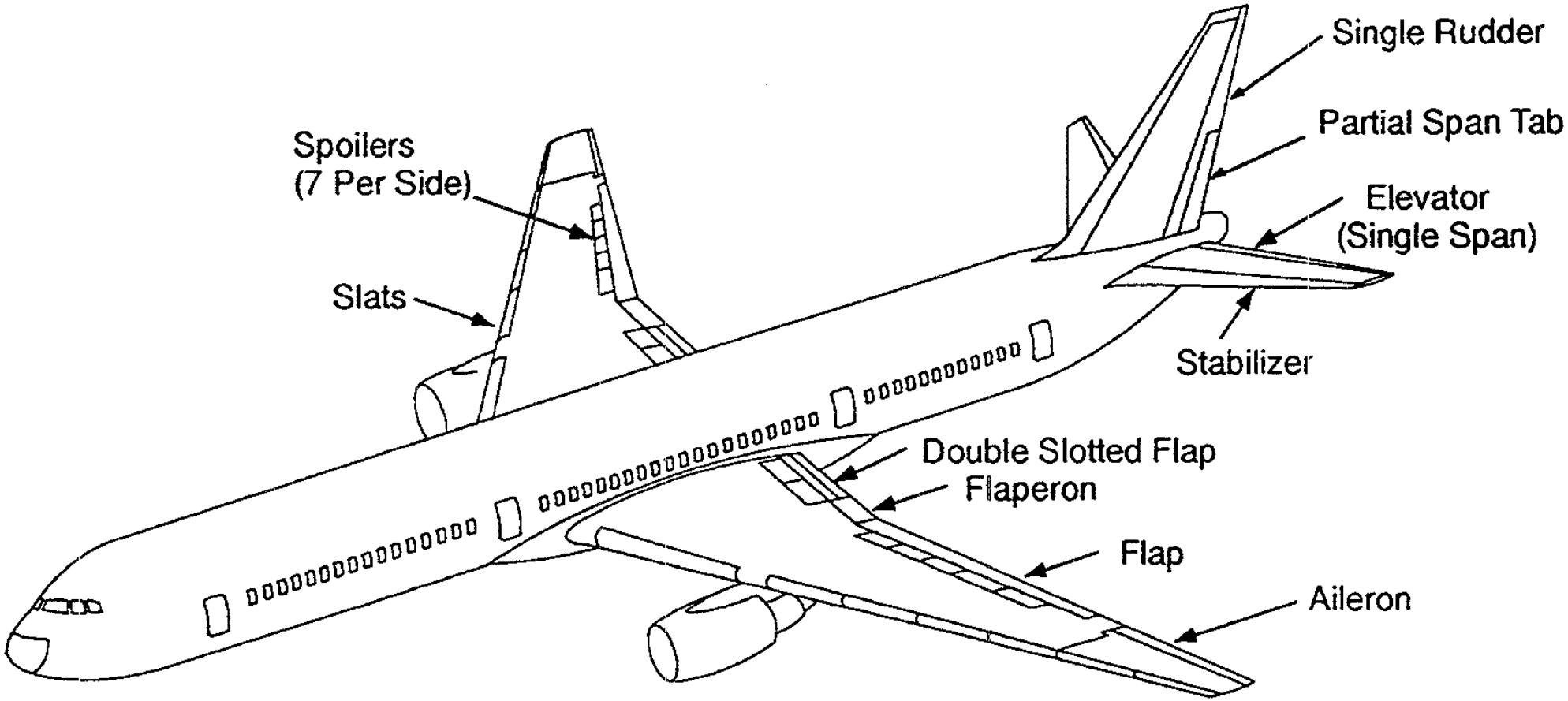
Introduction

- TMR is an example of hardware redundancy.
- Three modules independently compute three outputs from the same input. A majority voter combines the outputs into a single output.
  - All modules produce same output: Voter chooses the common output.
  - The output of one module differs from the other two: Voter chooses the output of the two agreeing modules.
  - All modules produce different output: Voter signals error.
- Able to correct one failing module and detect two failing modules.
- TMR is used for example in aerospace industry.
  - Radiation may cause hardware errors.



Flight control surfaces

Boeing 777

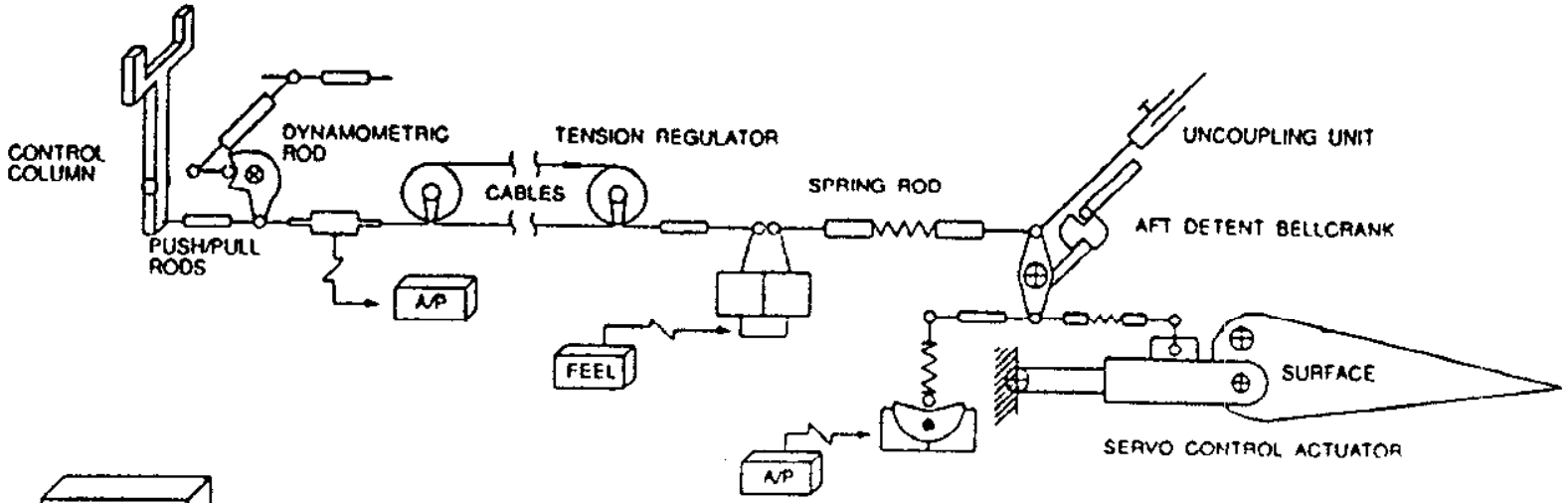


Source: Y. C. Yeh (Boeing), "Triple-Triple Redundant 777 Primary Flight Computer", Proc. Aerospace Applications Conference, 1996

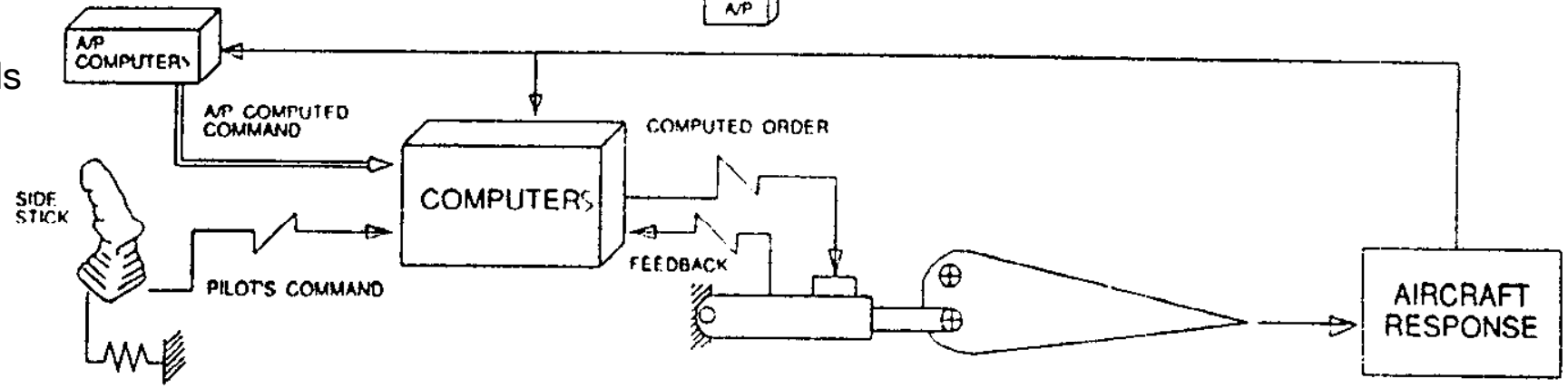
Flight control systems

Fly-by-wire

Conventional flight controls



Electrical flight controls



A/P: Autopilot

Source: D. Brière, P. Traverse (Aérospatiale), "AIRBUS A320/A330/A340 Electrical Flight Controls — A Family of Fault-Tolerant Systems", 23d Int'l Symp. on Fault-Tolerant Computing, 1993

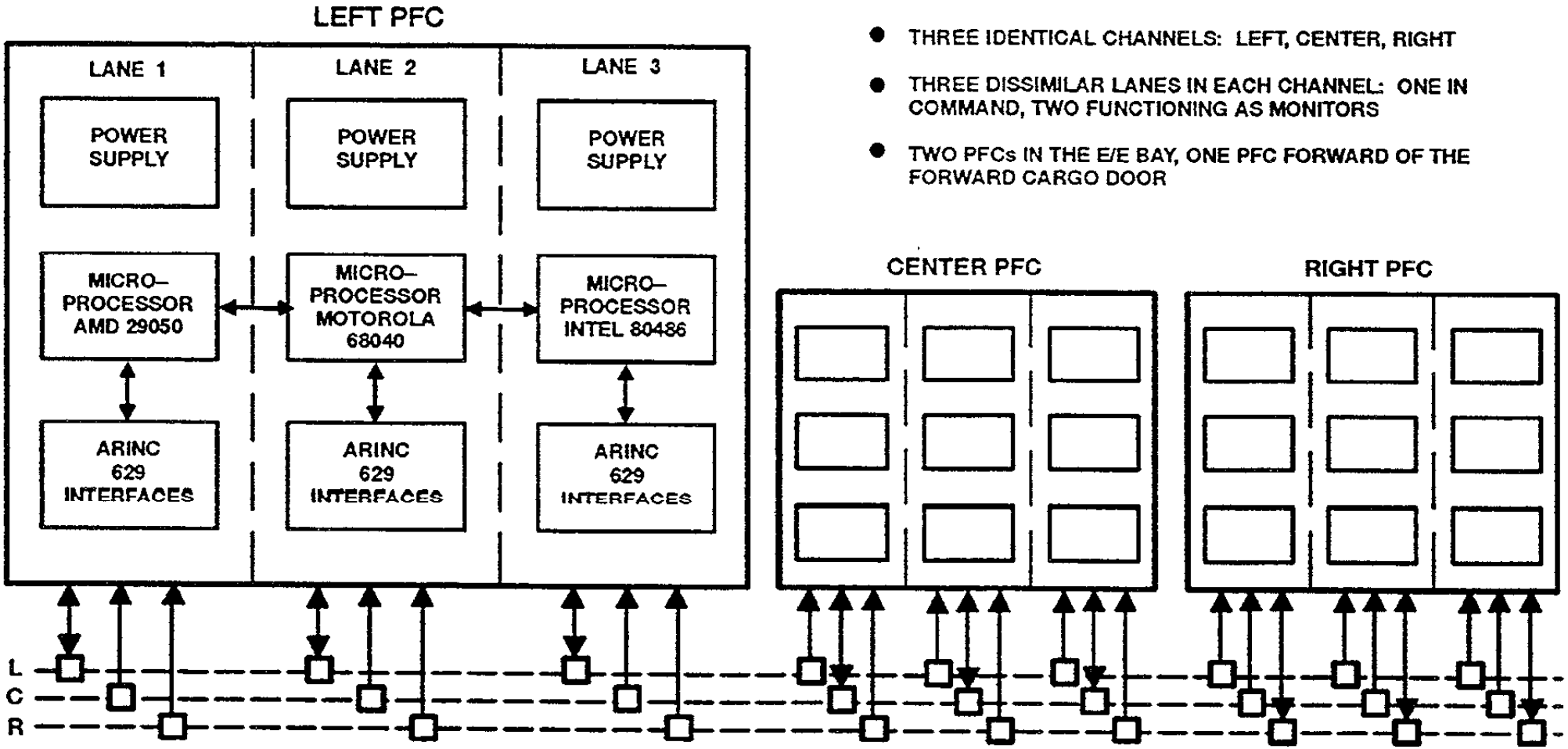
Boeing 777 redundancy principles

Primary Flight Control system

- **Triple redundancy** for all hardware resources:
  - Computing system
  - Airplane electrical power
  - Hydraulic power
  - Communication path
- **Median select:** Three lanes (= computers) within the Primary Flight Control system independently calculate a proposed surface command. The command lane will perform a median select of three inputs of each variable (as opposed to majority vote).
- **Triple-triple Primary Flight Control system**
  - One PFC has three independent lanes
  - Airplane has three independent PFCs
  - Enables delayed maintenance concept
- **Dissimilarity:** Design errors can defeat redundancy strategies, and can even result in shutdown of several computer channels. Therefore introduce dissimilarity in development process.
  - Dissimilar software development teams (not used in Boeing 777)
  - Dissimilar microprocessors
    - AMD 29050
    - Motorola 68040
    - Intel 80486
  - Dissimilar Ada compilers
  - Dissimilar hardware interfaces

Primary Flight Control (PFC) systems

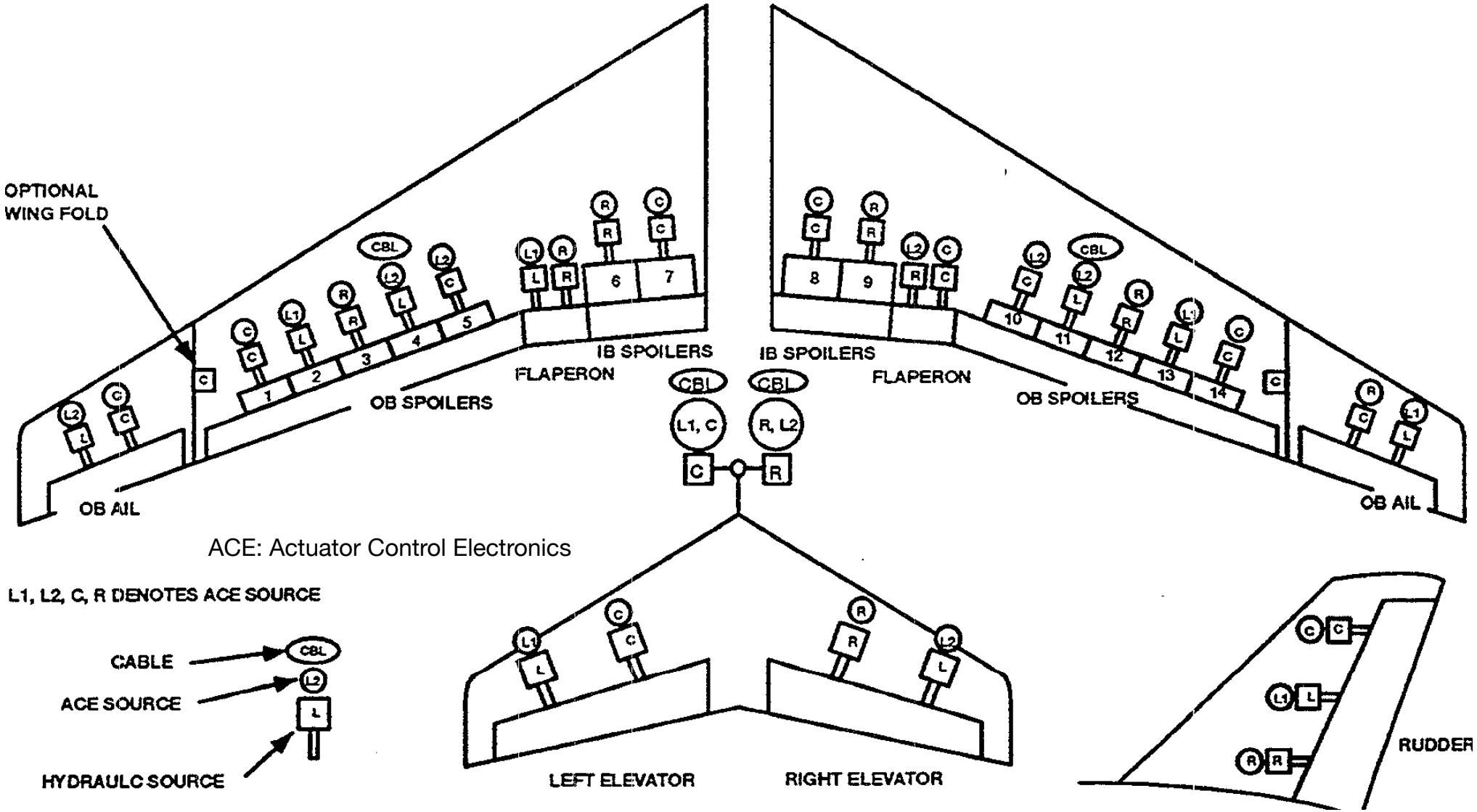
Triple-triple redundancy



- THREE IDENTICAL CHANNELS: LEFT, CENTER, RIGHT
- THREE DISSIMILAR LANES IN EACH CHANNEL: ONE IN COMMAND, TWO FUNCTIONING AS MONITORS
- TWO PFCs IN THE E/E BAY, ONE PFC FORWARD OF THE FORWARD CARGO DOOR

Actuator Control Electronics (ACE) units

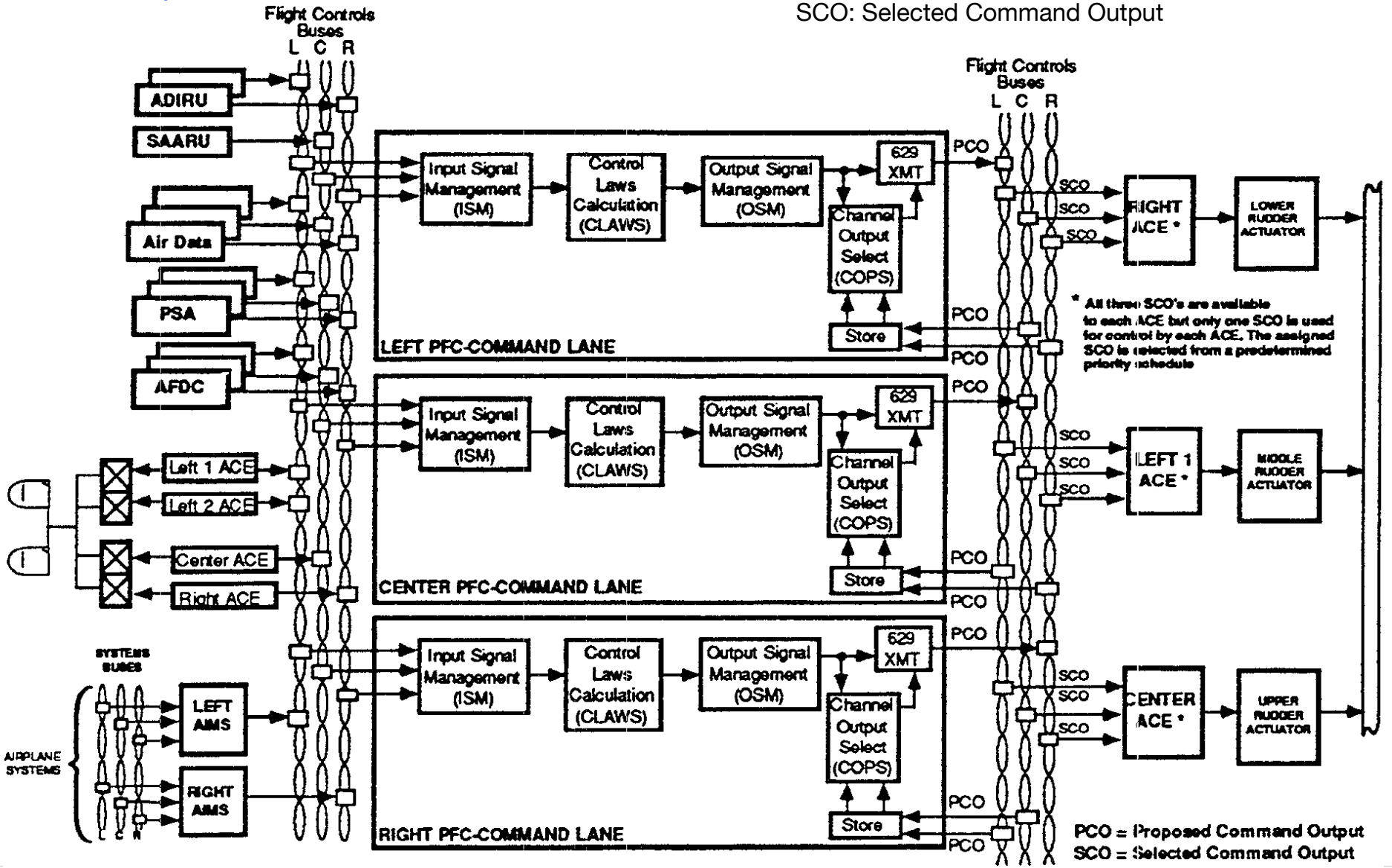
Boeing 777



ADIRU: Air Data Inertial Reference Unit
SAARU: Secondary Attitude and Air Data Reference Unit
PSA: Power Supply Assembly
AFDC: Autopilot Flight Director Computer
ACE: Actuator Control Electronics
PFC: Primary Flight Control
PCO: Proposed Command Output
SCO: Selected Command Output

PFC redundancy management

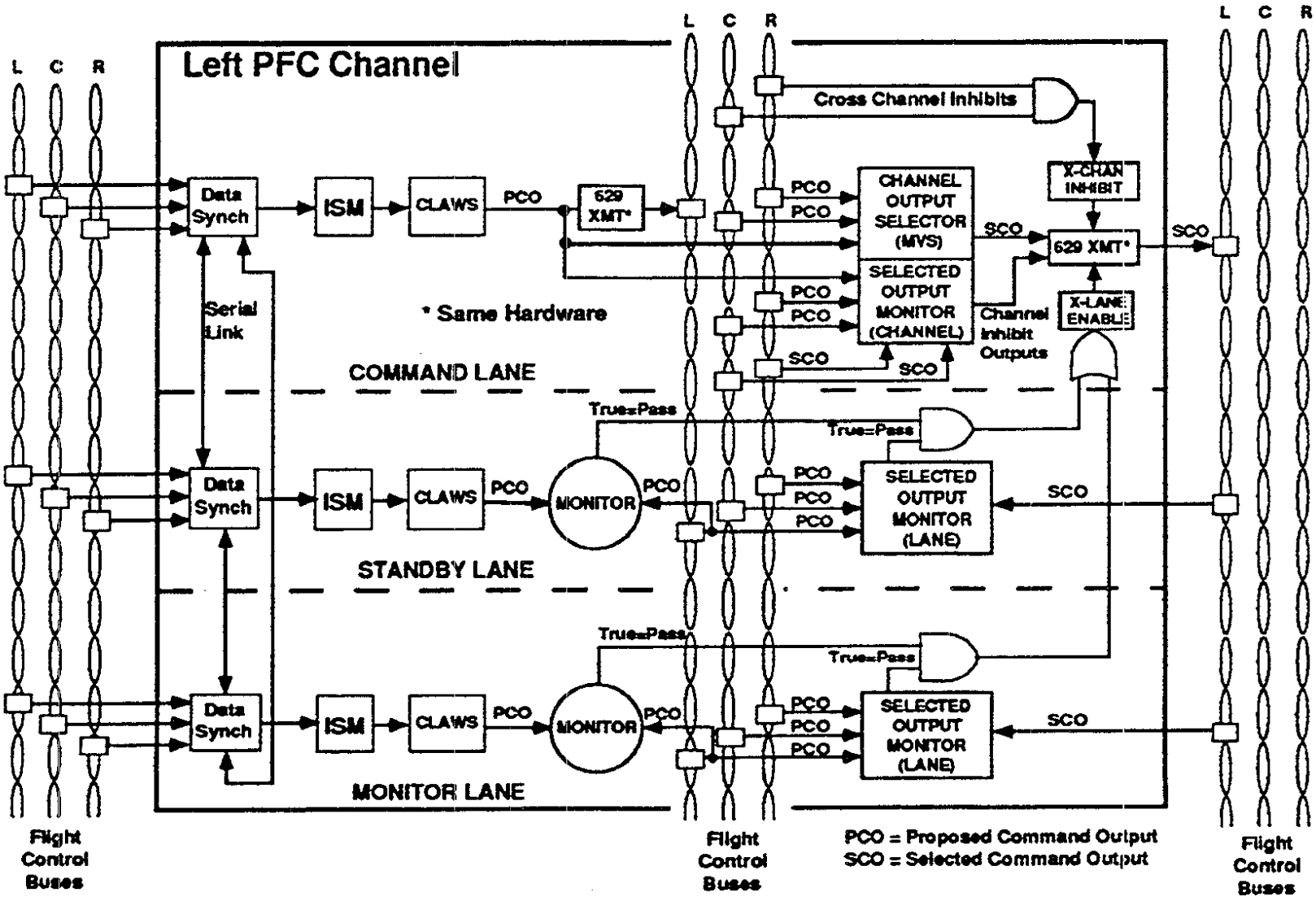
Typical control path



PFC redundancy management

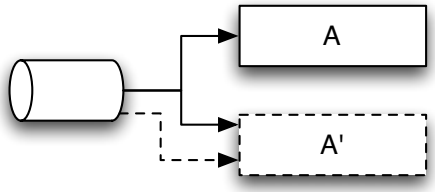
Output signal monitoring

ISM: Input Signal Management
CLAWS: Control Laws Calculation

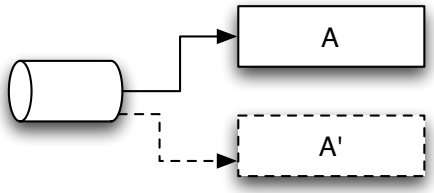


Failover
Introduction

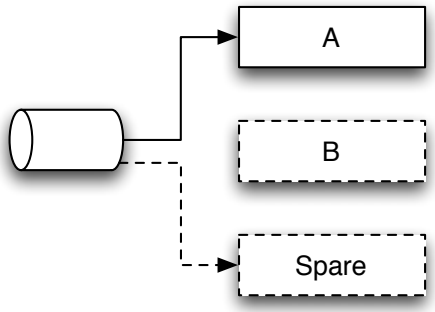
- **Failover** is an error recovery pattern.
  - System contains primary (active) element and a secondary (standby) element.
  - State of primary is replicated to secondary.
  - If primary fails, the secondary takes on the role of the active element → failover.
  - Needs *a person in charge* for steering the failover.
  - Needs proper *quarantine* for the faulty element.



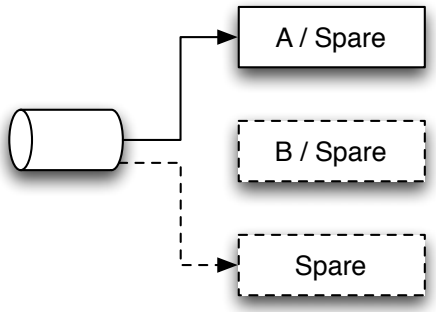
Active / Active



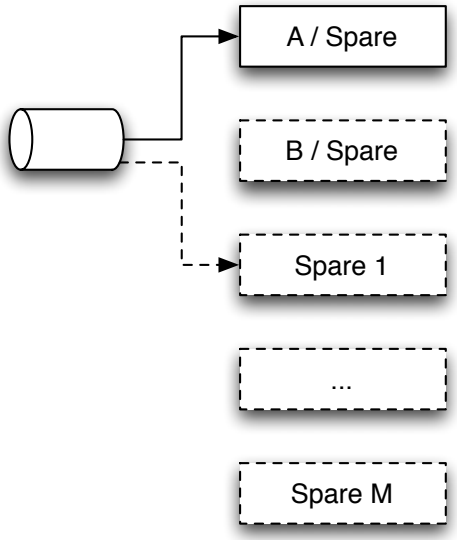
Active / Passive



N-to-1 (‚dedicated spare')



N+1 (‚roaming spare')



N + M

Failover

Redundancy configurations

- *N-to-1* and *N+1* are special cases of *Active / Passive* with multiple services
- *Active / Active* has no downtime, but leads to degraded system performance in failover case and might demand specialized data redundancy
- *N-to-1* has a dedicated spare and demands a fail-back step after repair. In contrast with *N+1* the elements switch roles and the fail-back is not needed.

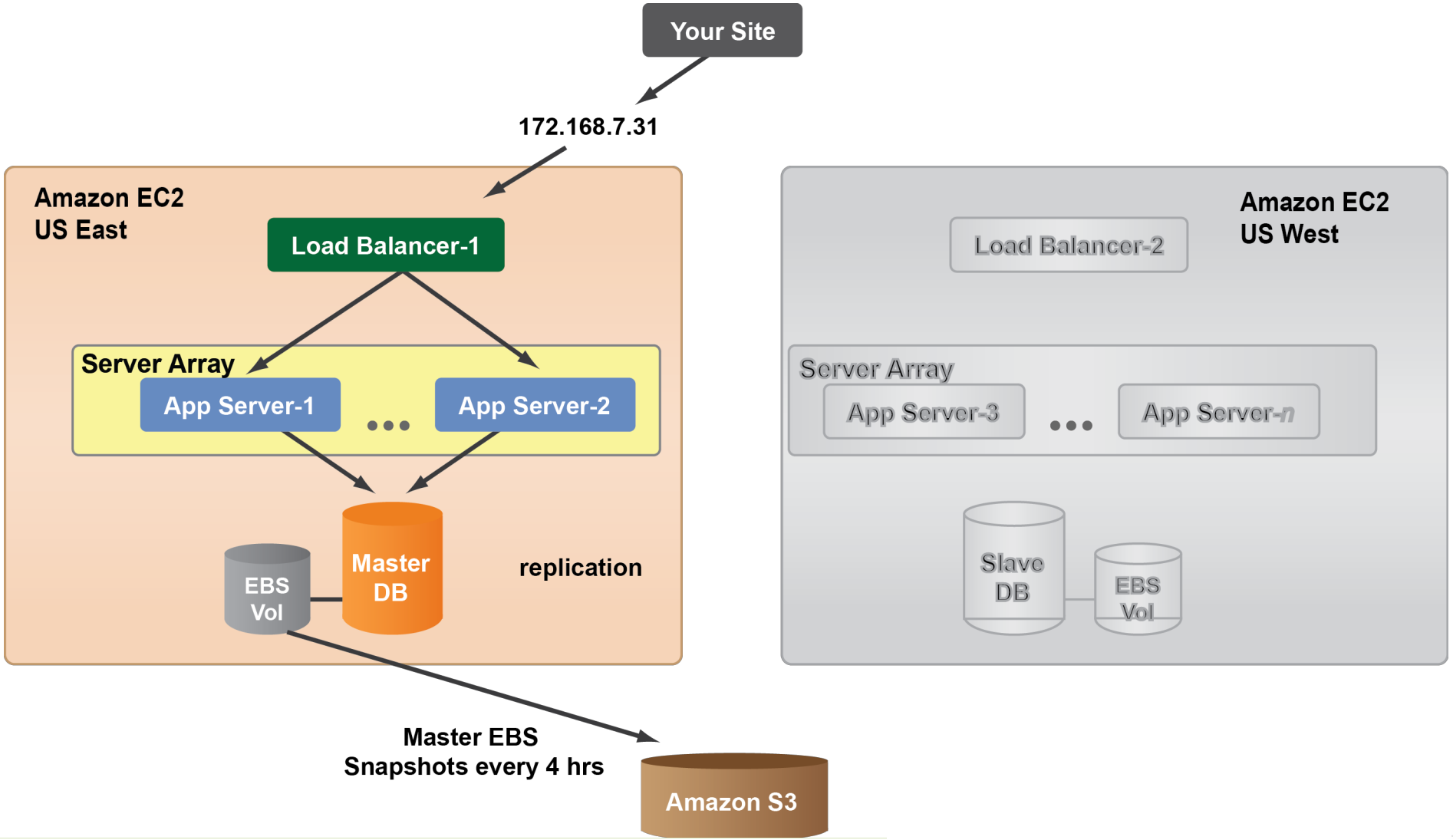
Failover

Cold / warm / hot standby

- **Cold standby**: The secondary is a blank, powered-off machine with hardware identical to the primary. In the case of a failure on the primary a system administrator powers on the secondary, installs and configures software, and restores data from the primary. Data from primary is backed up on a storage system and restored on secondary. This generally provides a **recovery time of a few hours**.
- **Warm standby**: The software components are installed and available on the secondary. The secondary is up and running. In the case of a failure on the primary, the software components are started on the secondary. This process is manual or automated using a cluster manager. Data is regularly mirrored to secondary using disk based replication or shared disk. This generally provides a **recovery time of a few minutes**.
- **Hot standby**: Software components are installed and available on both primary and secondary. The software components on the secondary are up but will not process data or requests. Data is mirrored in near real time and both systems will have identical data. Data replication is typically done through the software's capabilities. This generally provides a **recovery time of a few seconds**.
- **Active-Active** (Load balanced): In this method both the primary and secondary are active and processing requests in parallel. Data replication happens through software capabilities and is bi-directional. This generally provides a **recovery time that is instantaneous**.

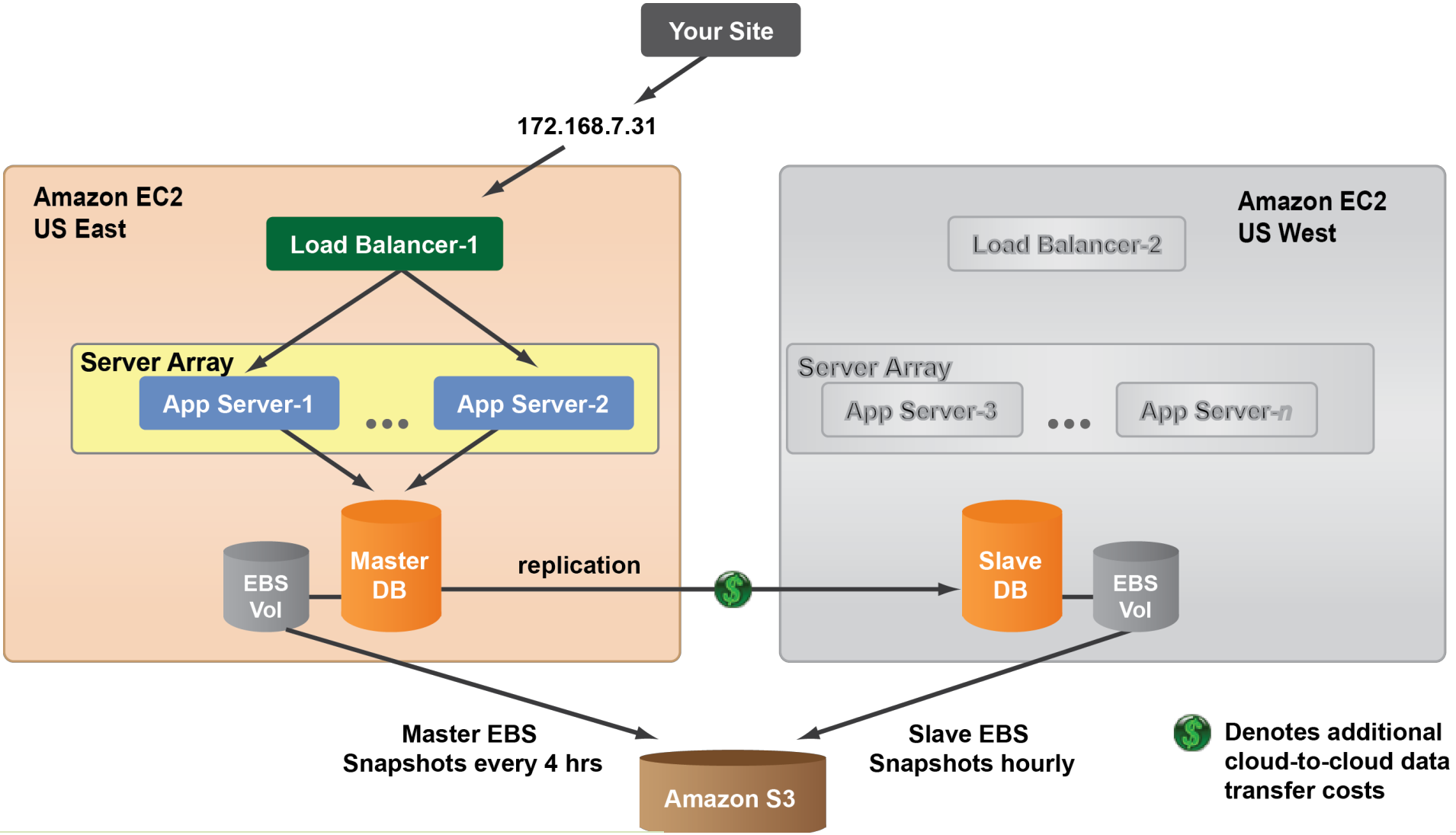
Failover in a High-Availability architecture

Example of cold standby



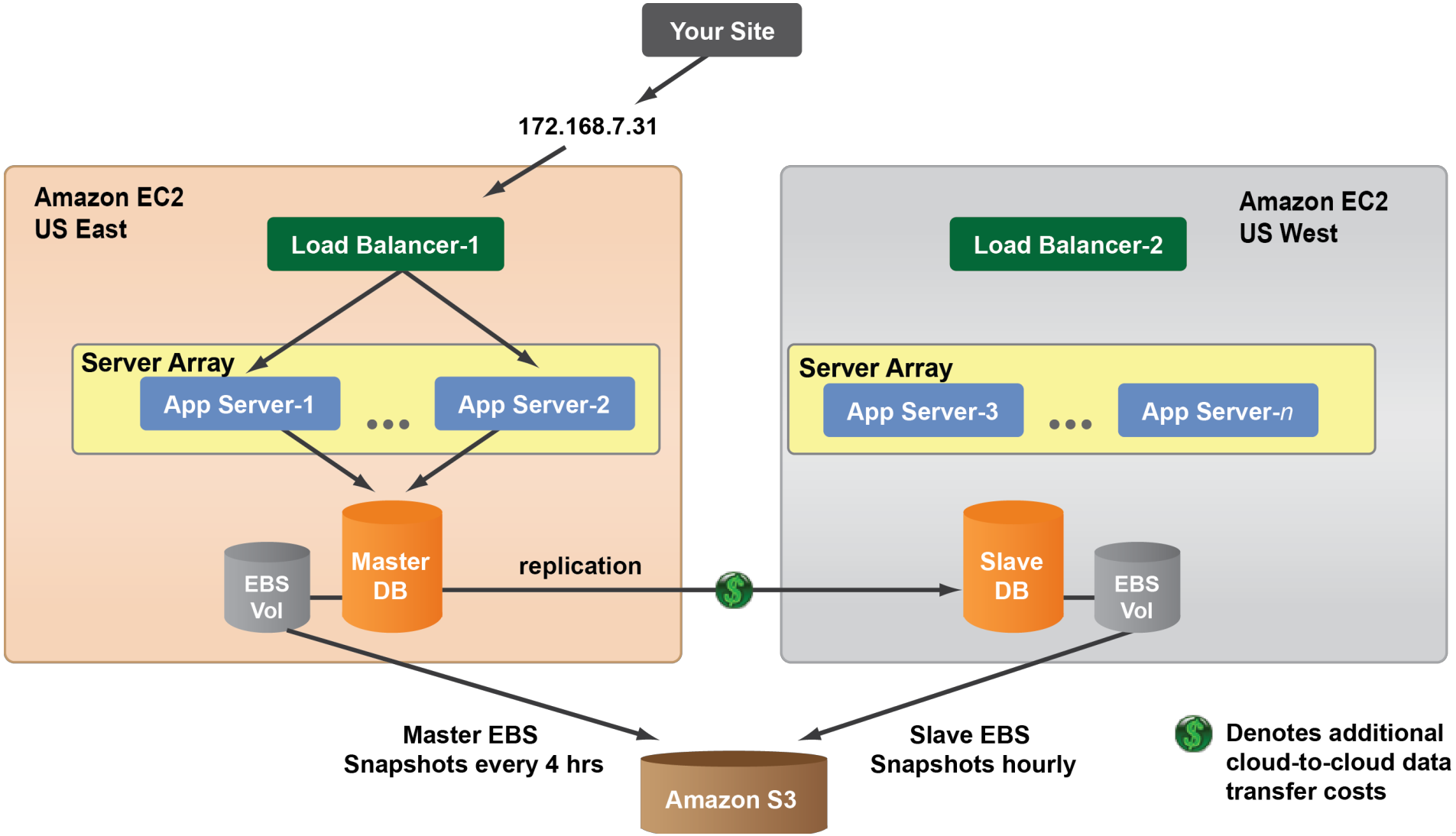
Failover in a High-Availability architecture

Example of warm standby



Failover in a High-Availability architecture

Example of hot standby



Failover in a High-Availability architecture

Example of active-active configuration