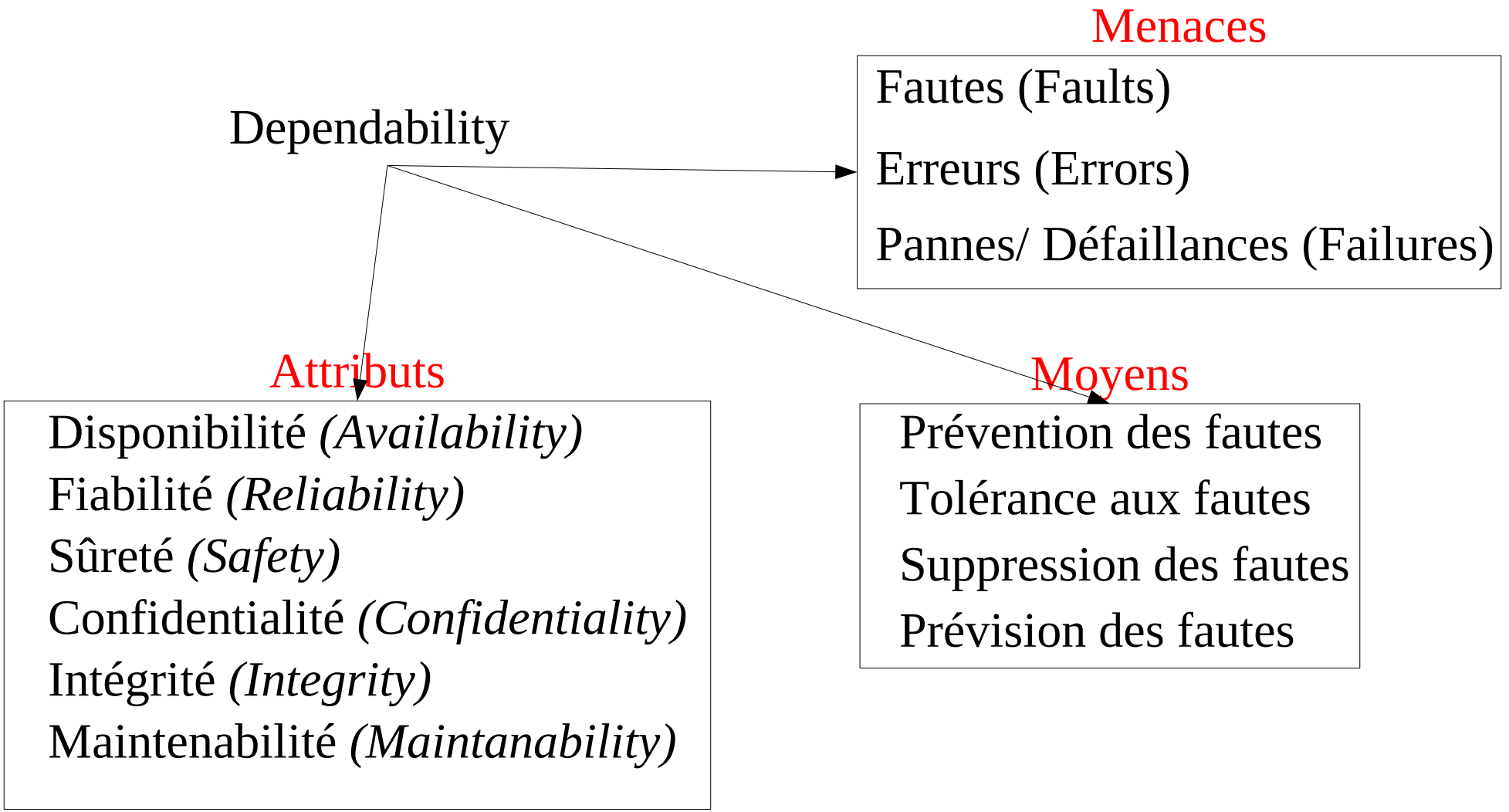
M2 SRI

Tolérance aux pannes

Plan

- Terminologie
- Modélisation
- Ex: Téléphonie
- Moyens locaux à un noeud
- Moyens distribués

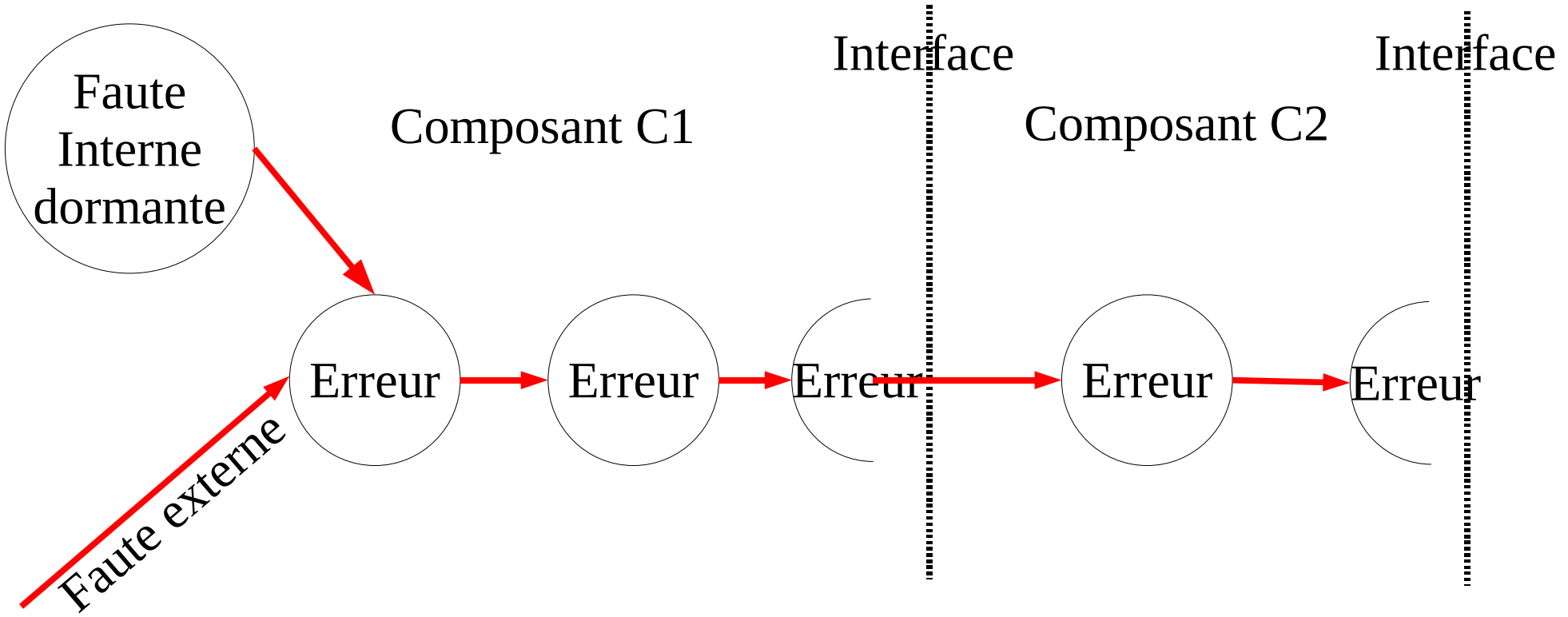
Dependability

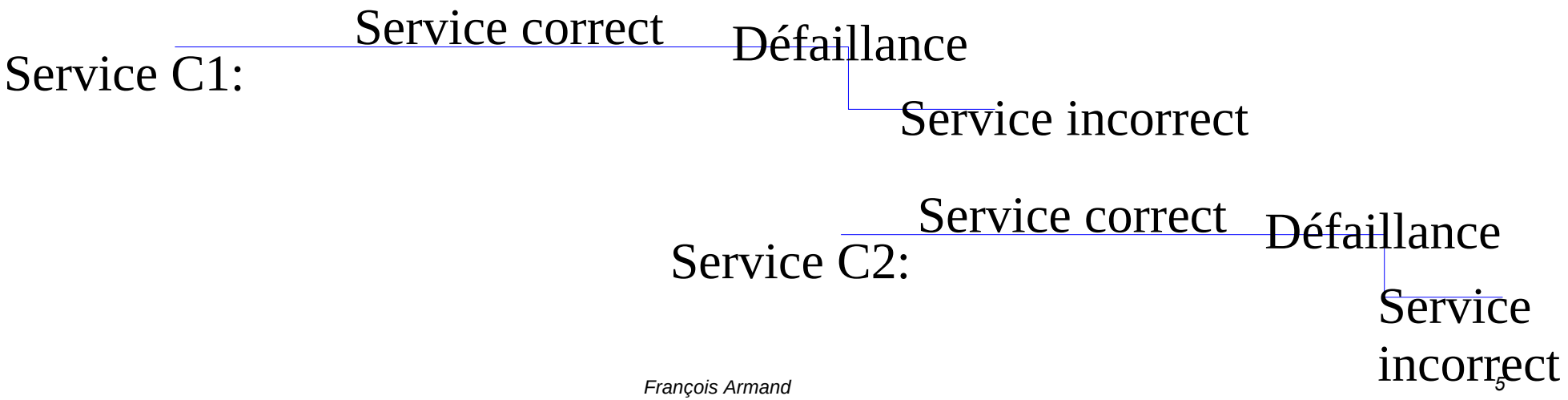


Menaces

- Faute:
  - Défaut physique du matériel ou du logiciel
- Erreur:
  - Une valeur incorrecte dans le système
- Défaillance / Panne:
  - Déviation du système par rapport à sa spécification

Menaces

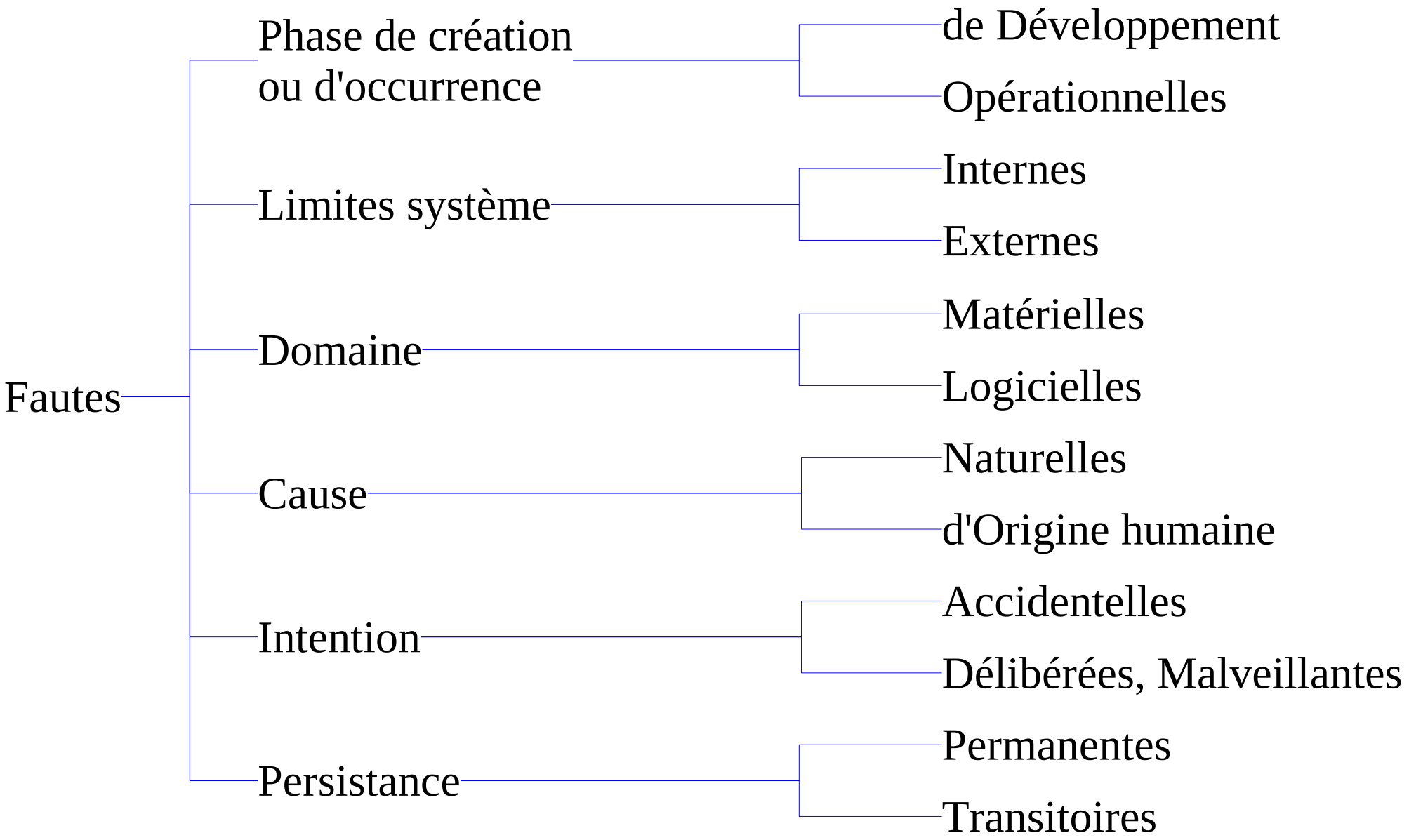

Faute
Interne
dormante
Composant C1
Interface
Composant C2
Interface
Erreur
Erreur
Erreur
Erreur
Erreur
Faute externe


Service C1:
Service correct
Défaillance
Service incorrect
Service C2:
Service correct
Défaillance
Service
incorrect

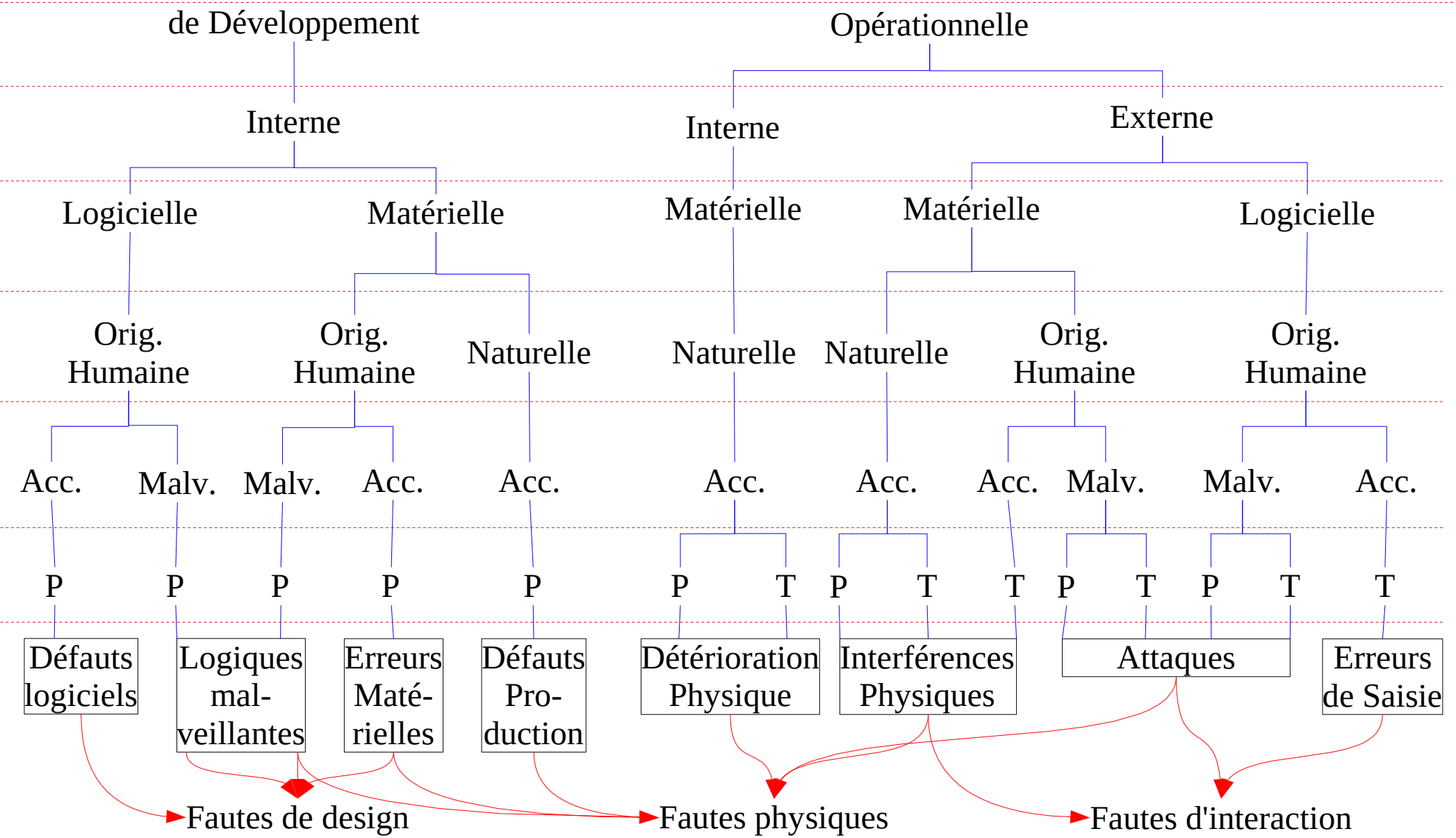
Modes de Défaillance

- Domaine:
  - Valeur erronée, défaillance temporelle
- Perception par plusieurs utilisateurs:
  - Défaillance cohérente ou incohérente
- Conséquences:
  - Mineures
  - ....
  - Catastrophiques

Classes Élémentaires de Fautes



Classes Combinées de Fautes



Attributs

- Disponibilité:
  - Prêt à fournir un service correct
- Fiabilité:
  - Fournir un service correct de manière continue
- Sûreté:
  - Absence de conséquences catastrophiques
- Confidentialité:
  - Ne pas dévoiler des informations sans autorisation

Attributs

- Intégrité:
  - Absence d'altérations incorrectes de l'état du système
- Maintenabilité:
  - Capacité à supporter des modifications et des réparations

Moyens

- Prévention:
  - Contrôle qualité
  - Méthodes de conception,
  - Programmation structurée,
  - Formation (des opérateurs,...)
  - Protection contre les radiations,
  - Firewalls,...
  - ....

Moyens

- Tolérance aux fautes:
  - Détection des erreurs
  - Recouvrement des erreurs
    - rollback
    - rollforward
  - Gestion des fautes:
    - Diagnostic,
    - Isolation
    - Reconfiguration
    - Ré-initialisation

Moyens

- Suppression:
  - Validation,
  - Vérification
    - statique,
    - dynamique,
    - Y compris les mécanismes de tolérance!
    - ==>Injection de fautes
  - Maintenance
    - Corrective
    - Préventive

Moyens

- Prévision:
  - Évaluation du comportement du système
  - Qualitatif
    - Identifier, classifier, analyser les pannes possibles
  - Quantitatif
    - Déterminer la probabilité avec laquelle le système satisfait aux attributs de dépendabilité.

Plan

- Terminologie
- Modélisation
- Ex: Téléphonie
- Moyens locaux à un noeud
- Moyens distribués

Disponibilité: Mesures

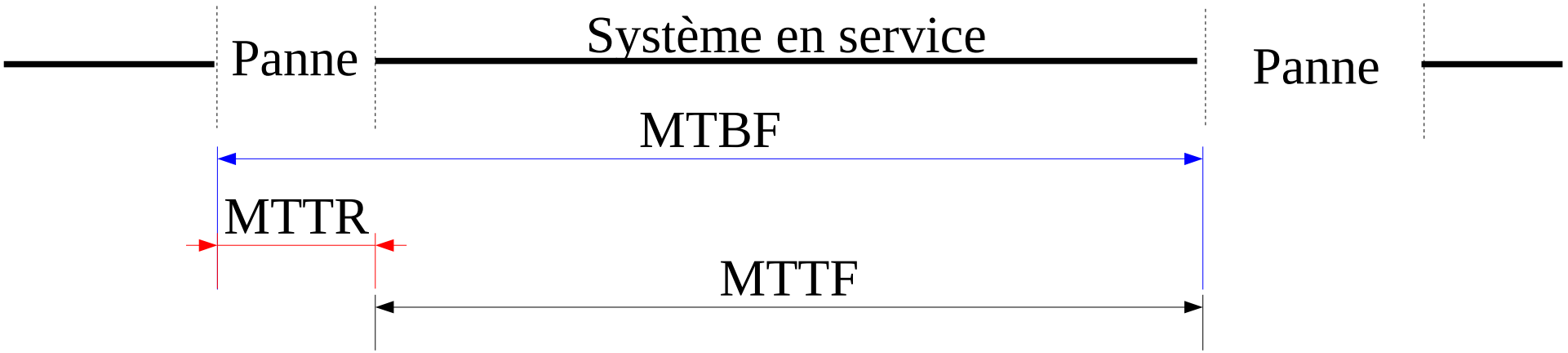
- Disponibilité:
  - Temps en service / Temps total
- Indisponibilité:
  - Temps hors service / temps total
- Disponibilité + Indisponibilité = 1
- Disponibilité exprimée en "nombre de neufs"
  - 4 '9': 0,9999 (ou 99,99 % du temps)
  - 5 '9': 0,99999 (ou 99,999 % du temps)

Disponibilité: Mesures

Classe	Type	Minutes indisponibilité par an	Disponibilité
1	Unmanaged	50.000: 34 jours, 17h 20mn	90%
2	Managed	5.000: 3 jours, 11h 20mn	99%
3	Well Managed	500: 8h 20mn	99,9%
4	Fault-Tolerance	50	99,99%
5	High-Availability	5	99,999%
6	Very High-Availability	0,5: 30 secondes	99,9999%
7	Ultra High-Availability	0,05: 3secondes	99,99999%

Temps entre pannes

- MTBF: Mean Time Between Failures
- MTTR: Mean Time To Repair



- MTTF: Mean Time To Failure (peu utilisé)

Temps de Réparation

- Pour le matériel:
  - Logistique:
    - Temps nécessaire pour recevoir le rapport de panne, trouver la pièce de rechange, et envoyer un réparateur et la pièce sur le site
  - Sur Site:
    - Temps nécessaire au réparateur pour effectuer la réparation

Taux de pannes

- Les taux de pannes se mesurent en 'FIT's
  - Failure In Time
  - 1 FIT = 1 faute pour un milliard d'heures 10^9
- On note les taux de pannes: λ
- λFIT = 10^9 / MTBF (en heures)
  - MTBF (en heures) = 10^9 / λFIT
- Si un système a un taux de pannes de 15000 FIT's
  - Il a un MTBF de 10^9 / 15000 (plus de 7 ans 1/2)

Taux de pannes

- Les taux de pannes s'additionnent!
  - Si un composant élémentaire a un un taux de pannes λ, un système combinant deux composants élémentaires a un taux de pannes 2λ!

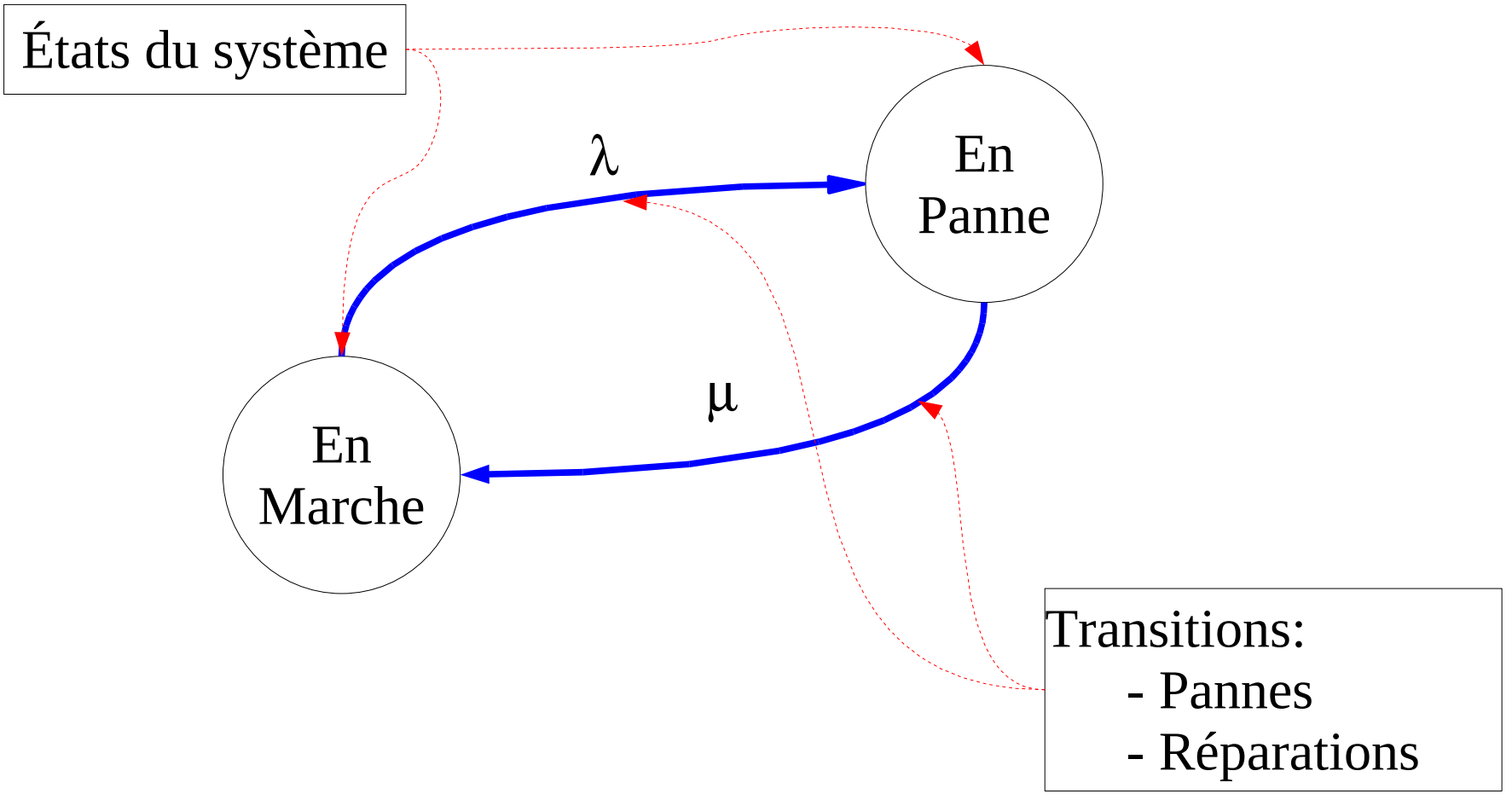
Taux de réparation

- Les taux de réparation:
  - se notent généralement μ
  - s'expriment aussi en FIT's
- Normalement,
  - $\mu >> \lambda$
  - Sinon, on va vers de sérieux problèmes!

Système Disponible

- Quand un système est-il disponible?
  - Un système est dit disponible, quand l'utilisateur dit qu'il l'est.
  - Les contraintes ne sont pas les mêmes pour tous les utilisateurs, une perte de performance de 10% suite à une panne peut être acceptable par un utilisateur et inacceptable par un autre.

Modèles de Markov



Modèles de Markov

- On cherche:
  - A déterminer l'occupation de chaque état du modèle: quelle fraction du temps est passée dans chacun de ses états.
  - L'occupation d'un état 'j' est $0 < P_j < 1$
  - La somme de l'occupation de tous les états du modèle est égale à 1: le système est forcément toujours dans l'un des états du modèle.

Modèles de Markov

- États: S_1 à S_j
- Occupation: P_j
- Transition de S_j à S_k est L_{jk}
- Taux de transition absolu sur une transition est: $P_j L_{jk}$
- Taux de transitions absolus entrants égaux aux taux de transitions absolus sortants

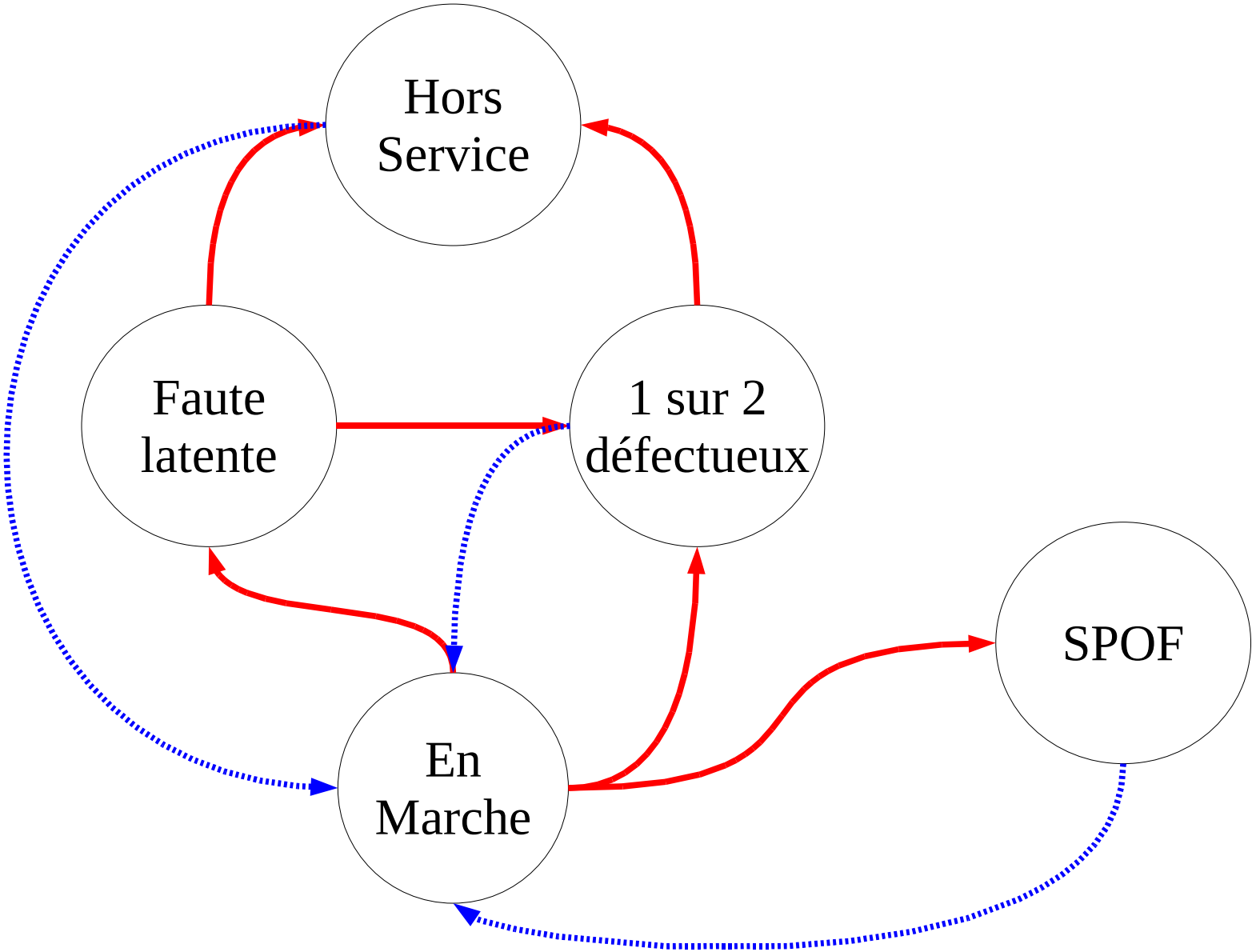
Modèles de Markov

- Il "suffit" de résoudre le système d'équations suivant:
- $\Sigma_{k=1,i} P_k L_{kj} = \Sigma_{k=1,i} P_j L_{jk} \forall\ j$
- $\Sigma_{k=1,i} P_k = 1$

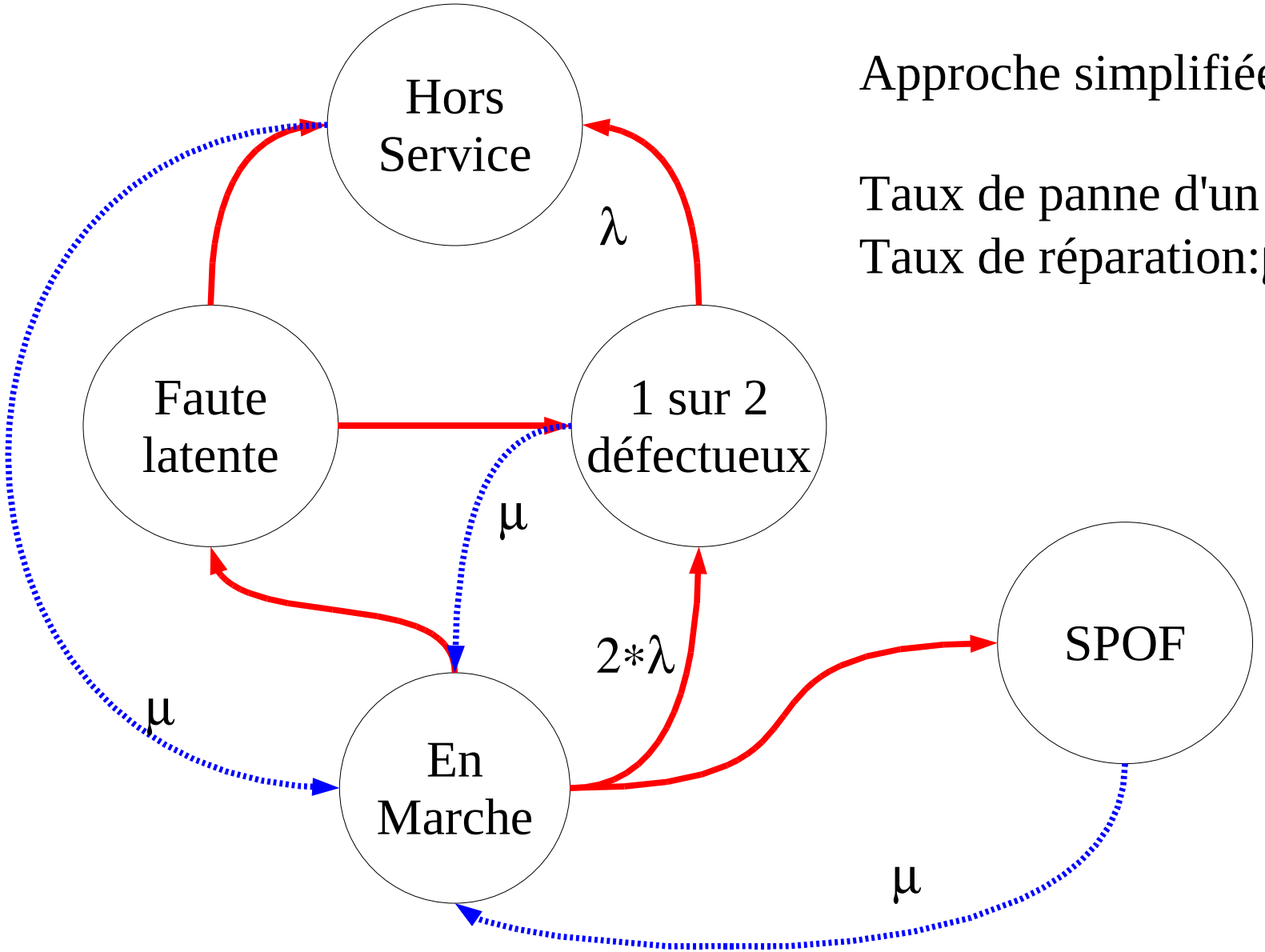
Fautes Latentes

- Faute non encore détectée qui conduira à la défaillance du système si elle est activée
- Un test de faute latente est une opération délibérée pour détecter ces fautes et en faire de vraies fautes détectées. Ces tests ne doivent pas causer la panne du système.

Exemple: Système redondant



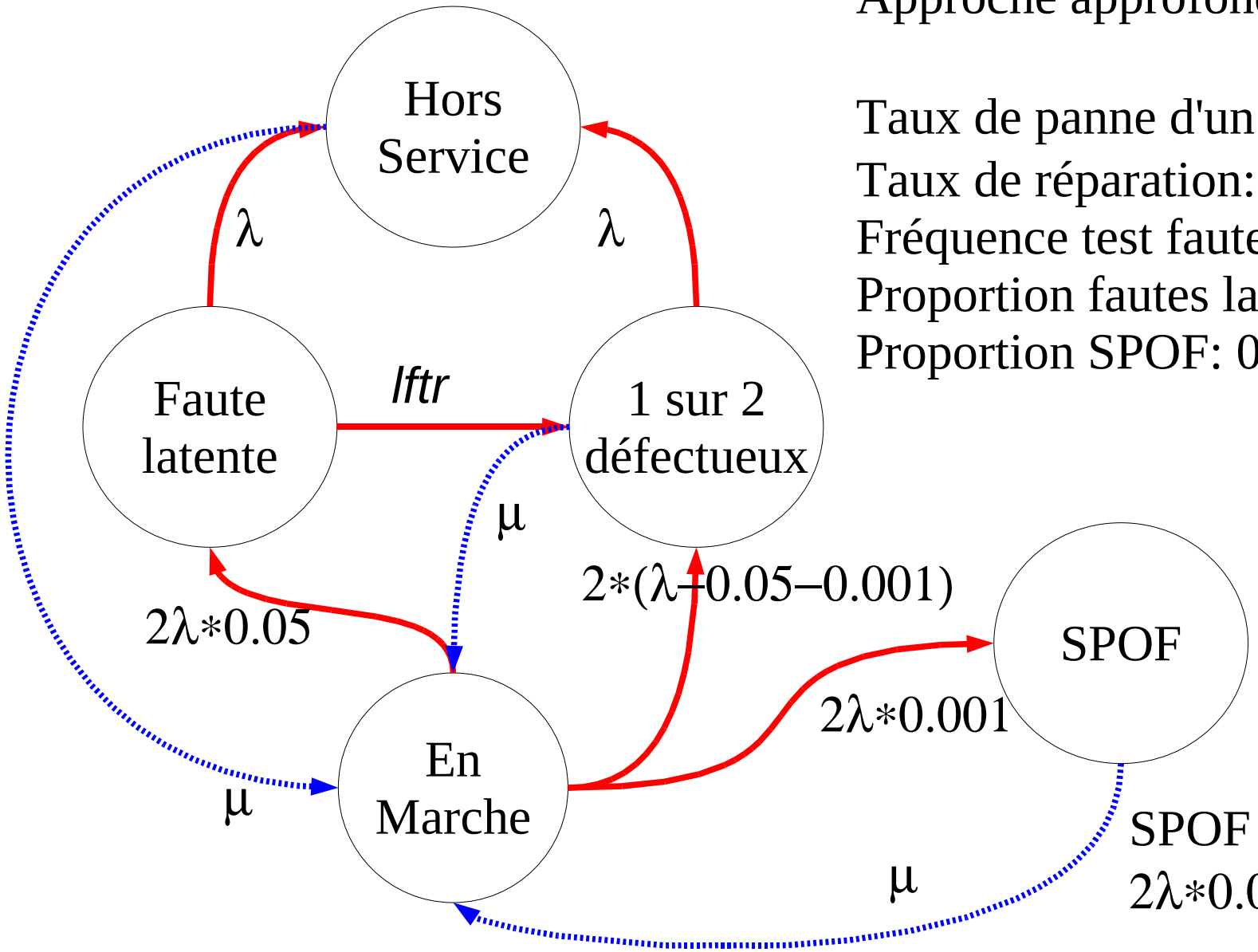
Exemple: Système redondant



Approche simplifiée:

Taux de panne d'un module:λ
Taux de réparation:μ

Exemple: Système redondant

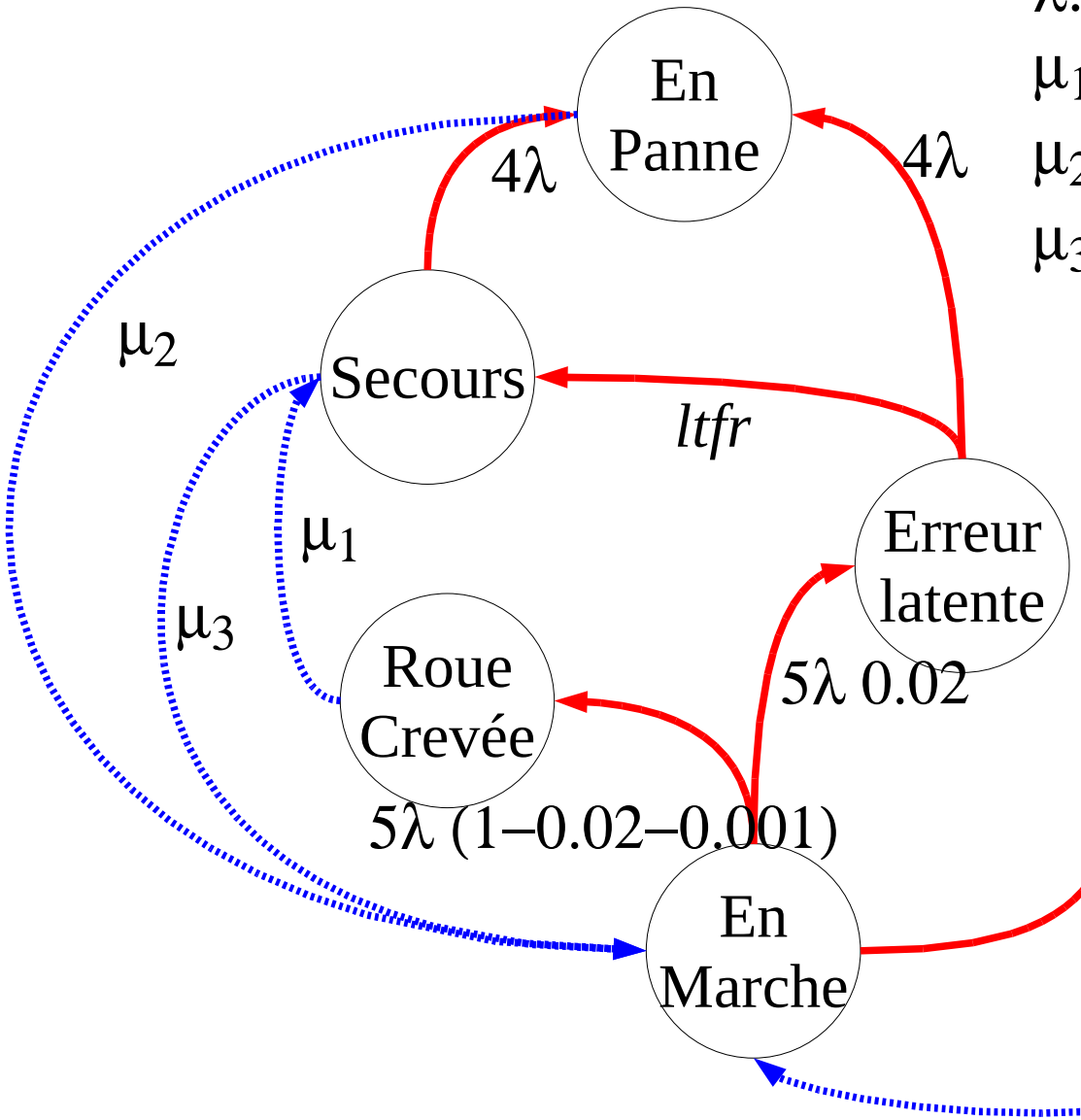


Approche approfondie:

Taux de panne d'un module:λ
Taux de réparation:μ
Fréquence test fautes latentes: *lftr*
Proportion fautes latentes: 0.05
Proportion SPOF: 0.001

SPOF transition:
2λ*0.001 ou λ*0.001

Exemple: Pneus d'une voiture



λ: 100 000 heures

μ_1: 10 mns changer roue

μ_2: 3 heures, appeler secours

μ_3: 1 semaine réparer roue crevée

États "système disponible" :

- En Marche
- Erreur latente
- Secours

En Pratique

- Modéliser composants bas niveau
  - Ex: disque, mémoire, hardware
- Les intégrer dans modèle plus large
- Ne pas oublier les "SPOF"
- Bien penser aux mécanismes de réparation

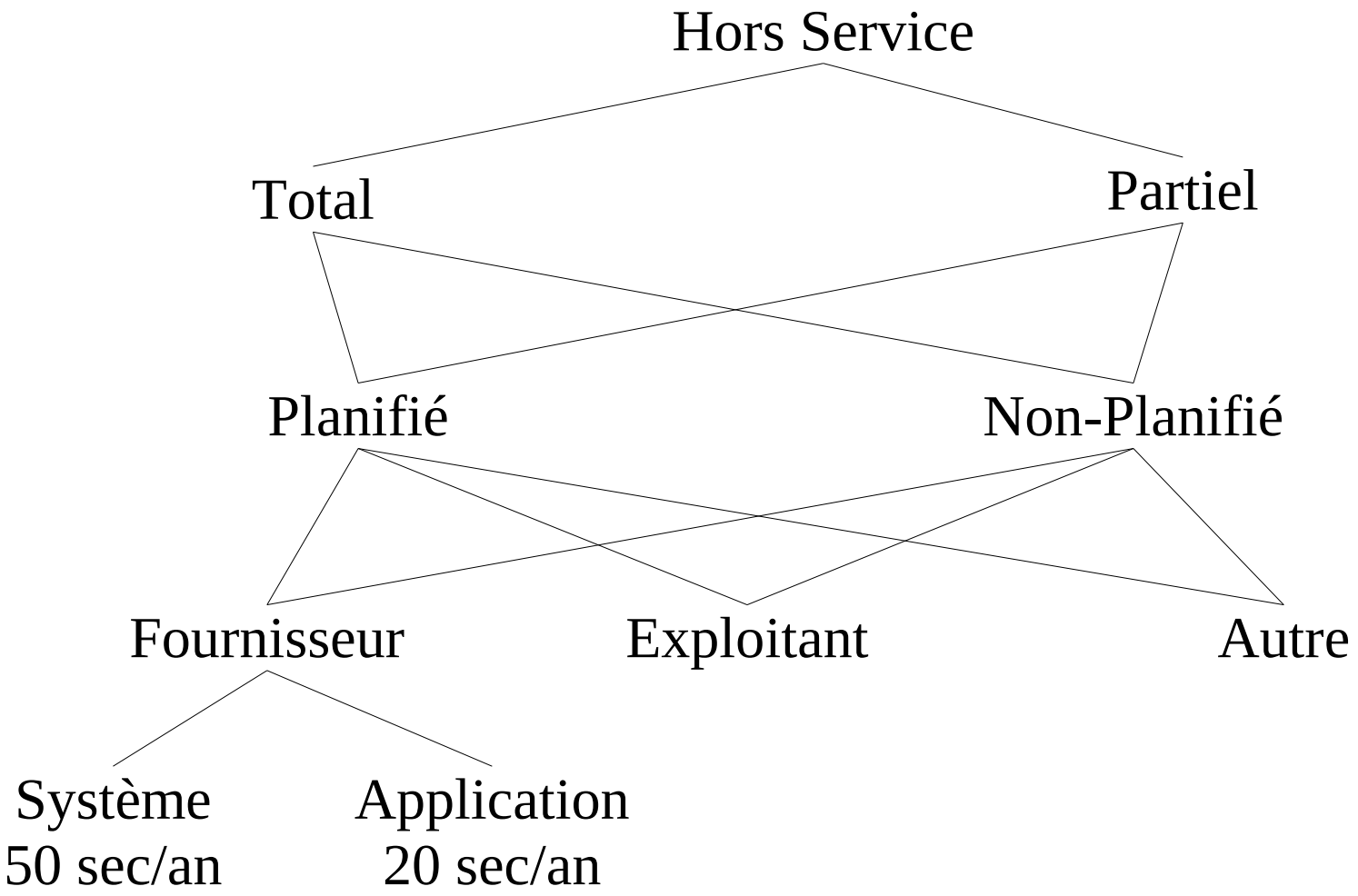
Plan

- Terminologie
- Modélisation
- Ex: Téléphonie
- Moyens locaux à un noeud
- Moyens distribués

Téléphone: Causes de Pannes

- Erreurs Humaines (Exploitant): 25%
- Erreurs Humaines Autres: 24%
- Erreurs Matérielles: 19%
- Erreurs Logicielles: 14%
- Nature: 11%
- Surcharge: 6%
- Vandalisme: 1%

Catégories de Pannes



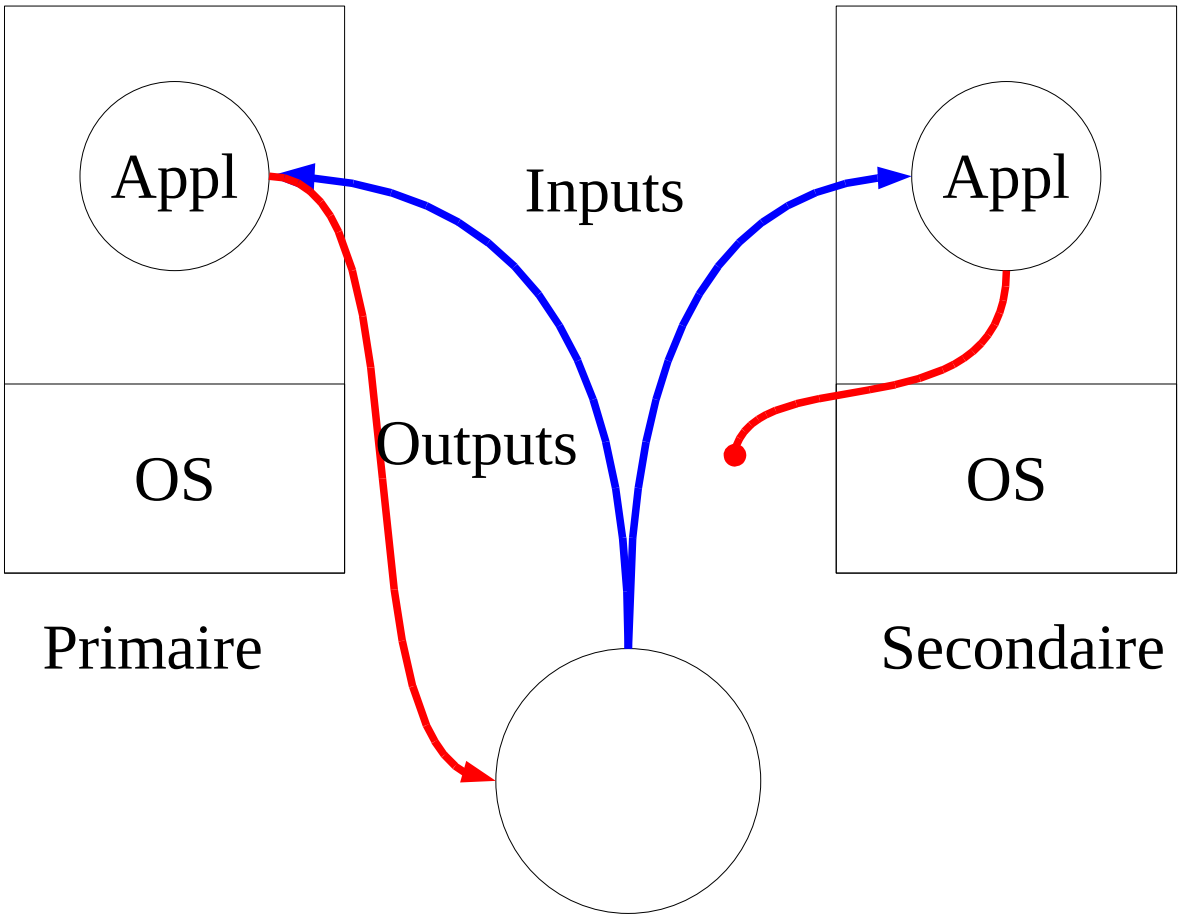
Budget de Disponibilité

- Panne Matérielle : 5 sec
- Défaillance système de recouvrement : 20 sec
- Erreurs Opérationnelles : 10 sec
- Défaillance logicielle système : 5 sec
- Défaillance Application: 30 sec
- TOTAL: 70 sec
  - Équivalent à 99,9997%

Types de redondances

- Hot standby
- Warm standby
  - 2N
  - 2N Load sharing
  - M+K (cas particulier: M+1)
- Spare

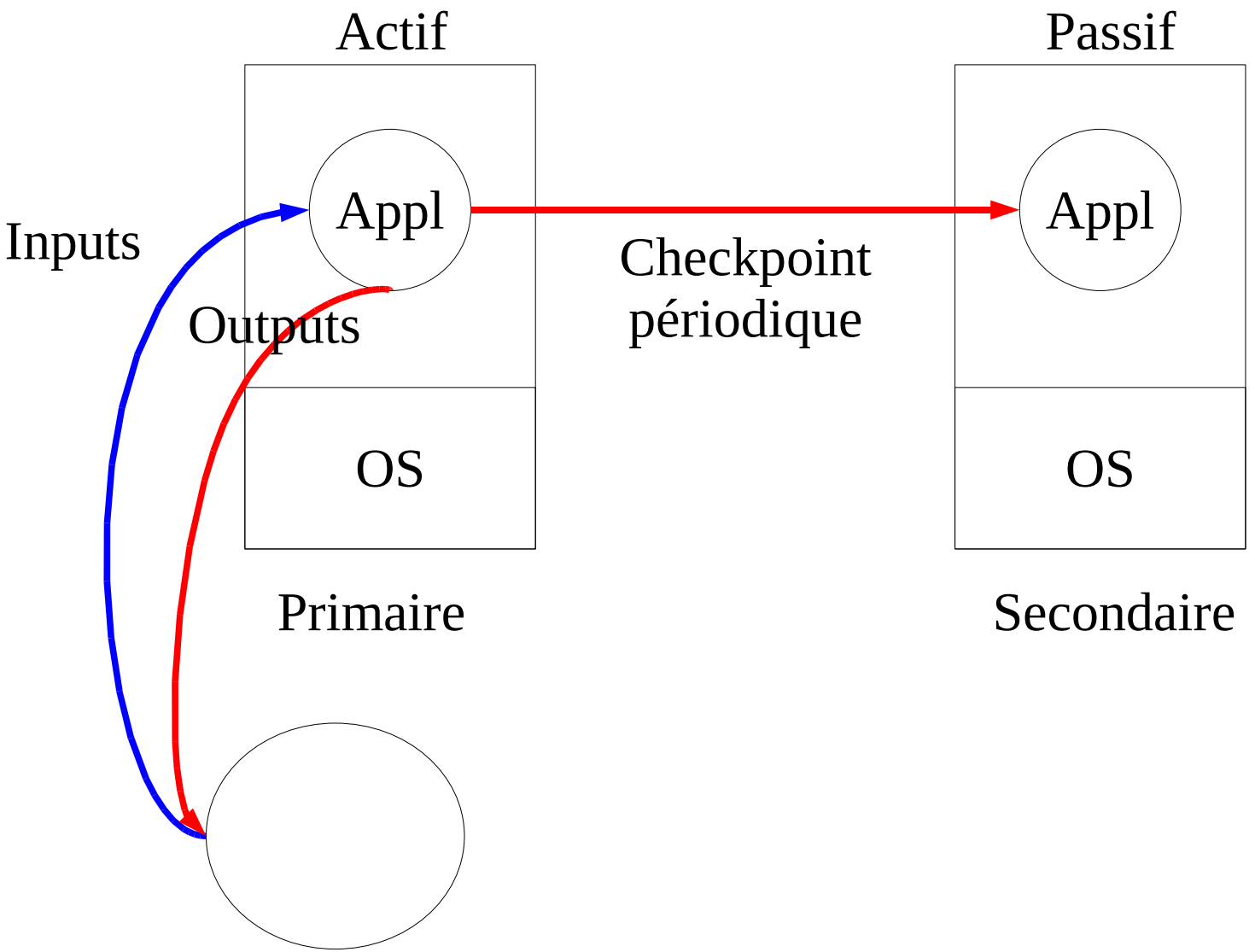
Hot Standby



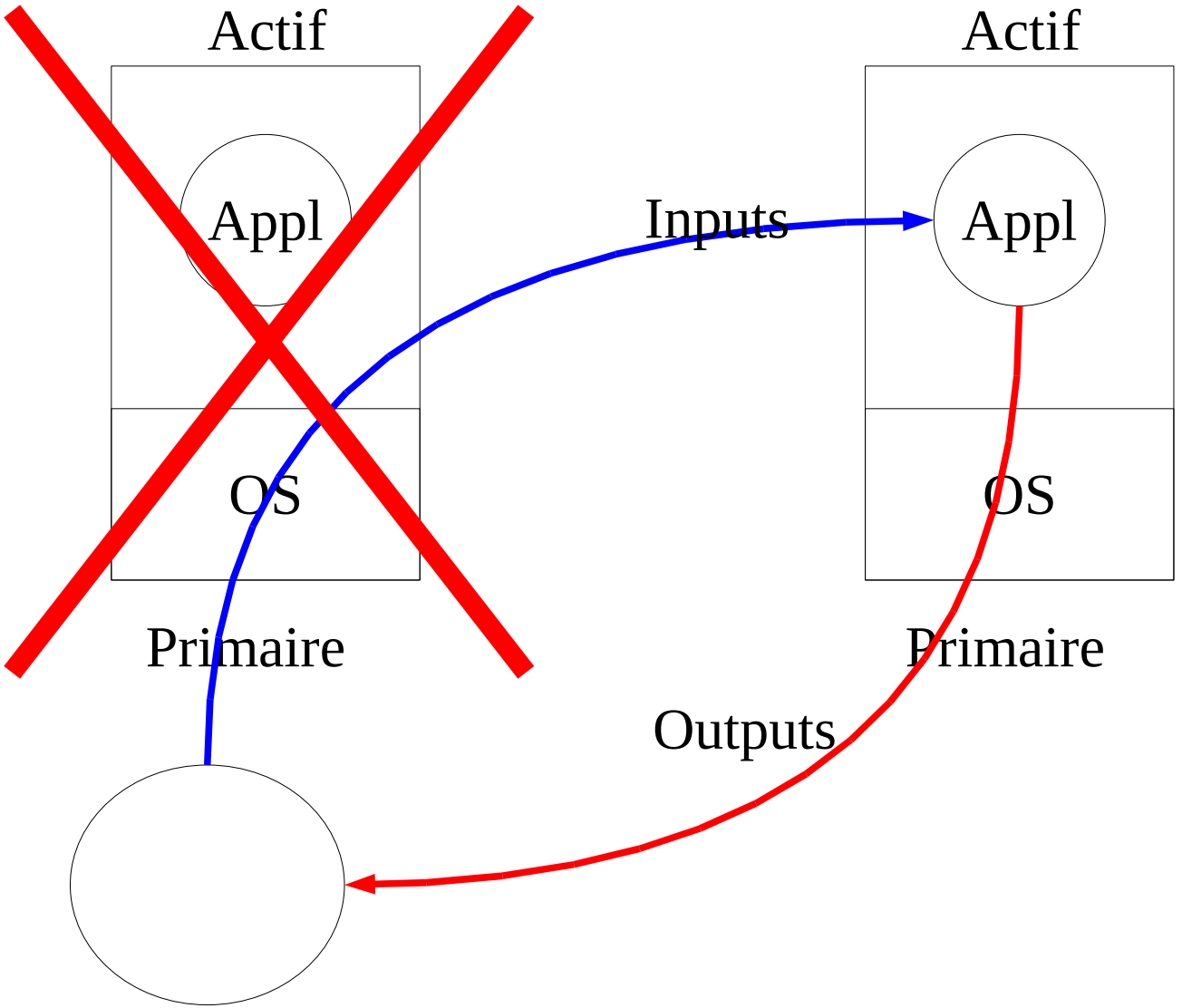
Hot Standby

- Nécessite le double des ressources
- Primaire et secondaire doivent se synchroniser
  - Difficile si applications utilisent mémoire partagée
- Failover: immédiat
- Réintégration difficile de la machine défaillante
- Protection contre les défaillances matérielles
- Même logiciel, mêmes conditions d'exécution
  - Mêmes défaillances!

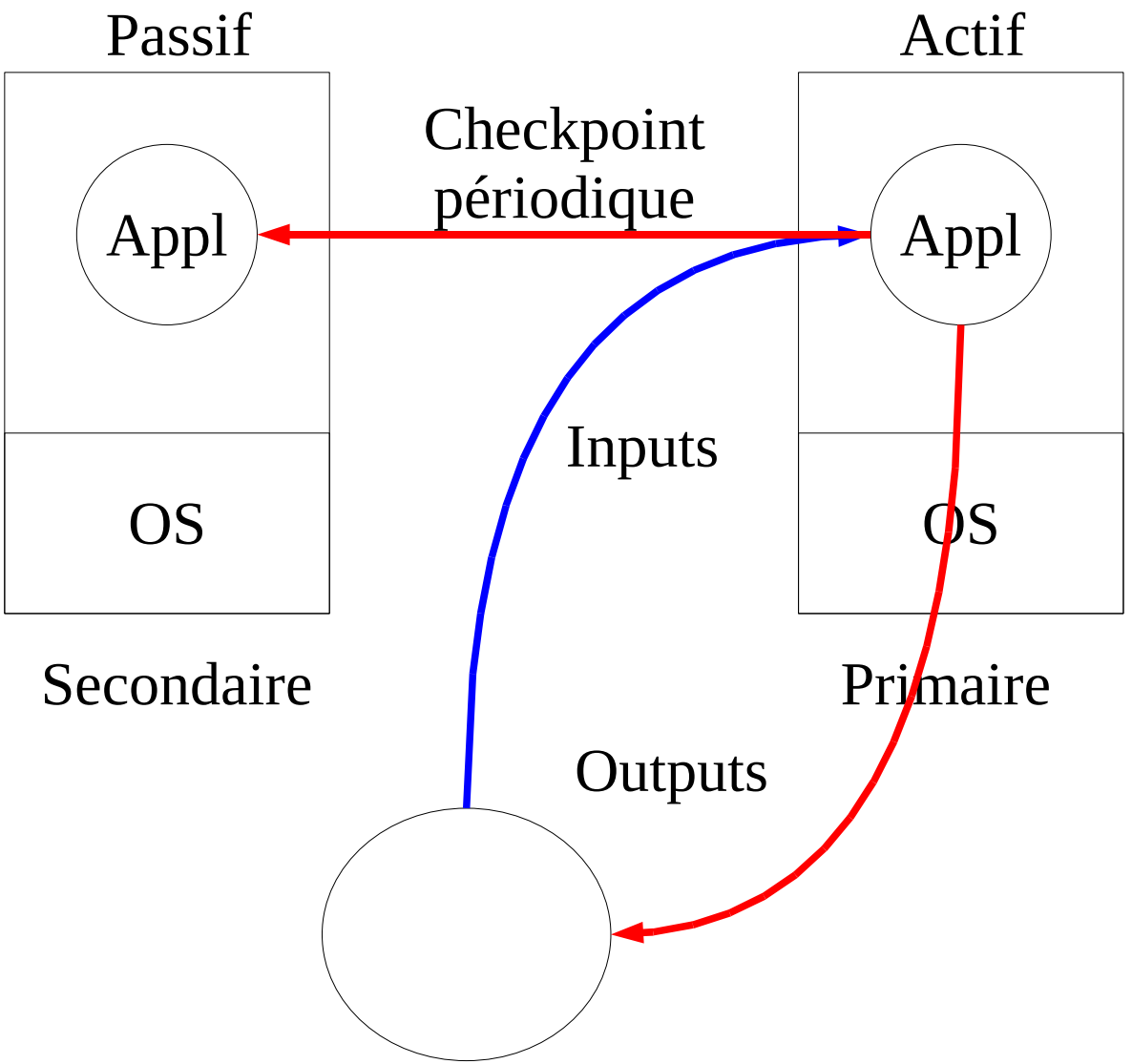
Warm standby: 2N



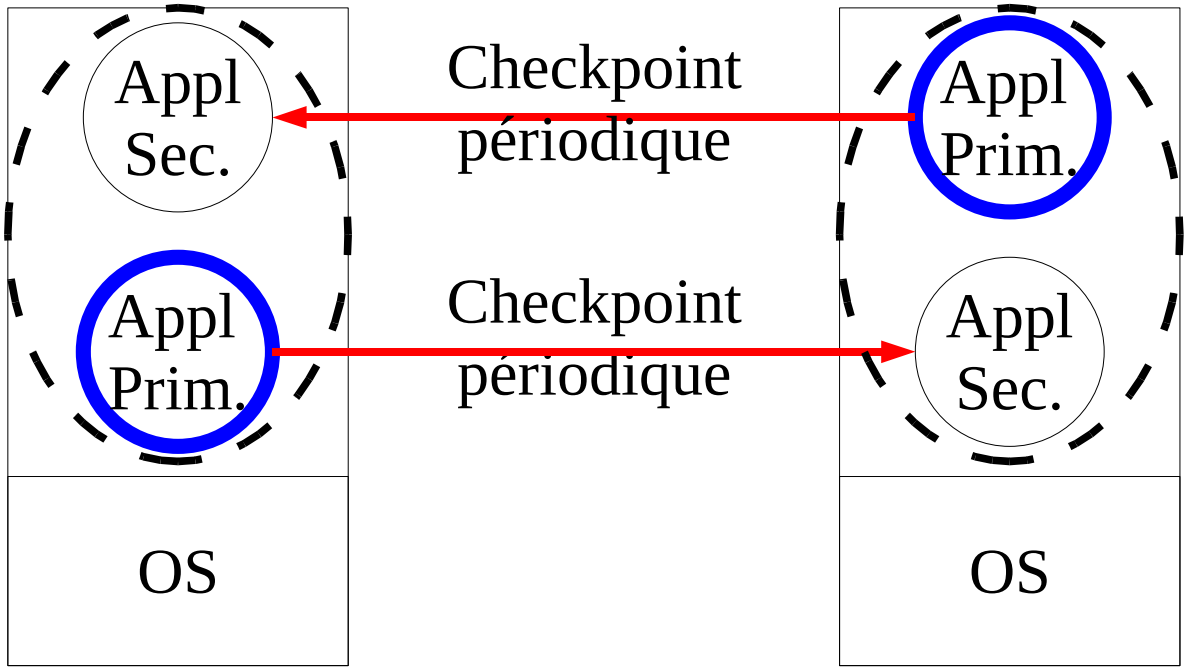
Warm standby: 2N



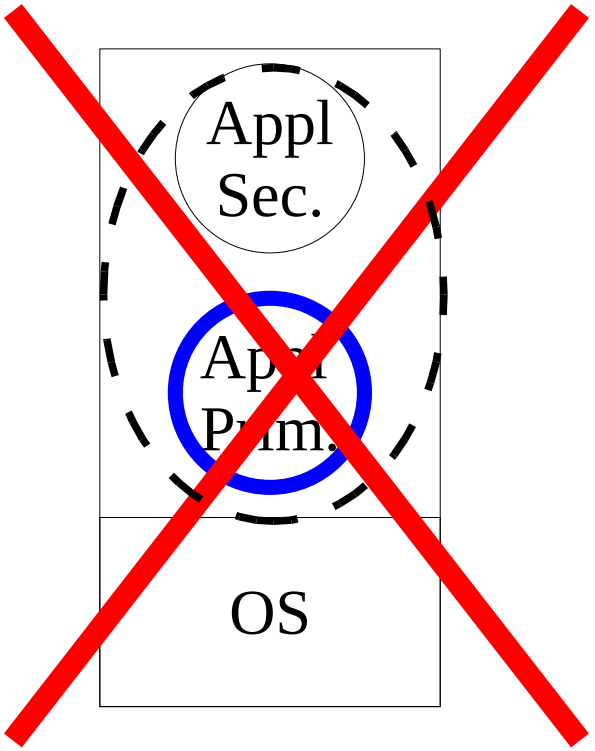
Warm standby: 2N

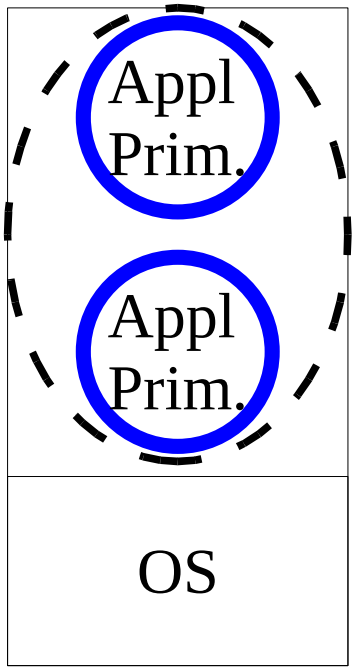


Warm standby: 2N / Load sharing

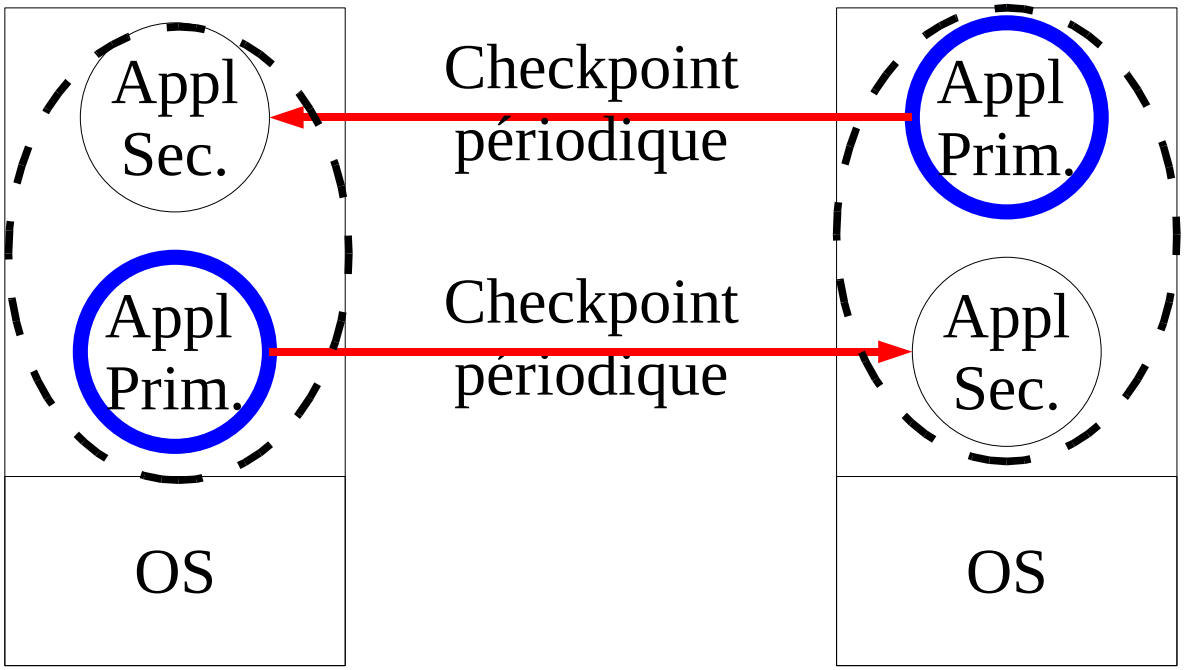


Warm standby: 2N / Load sharing

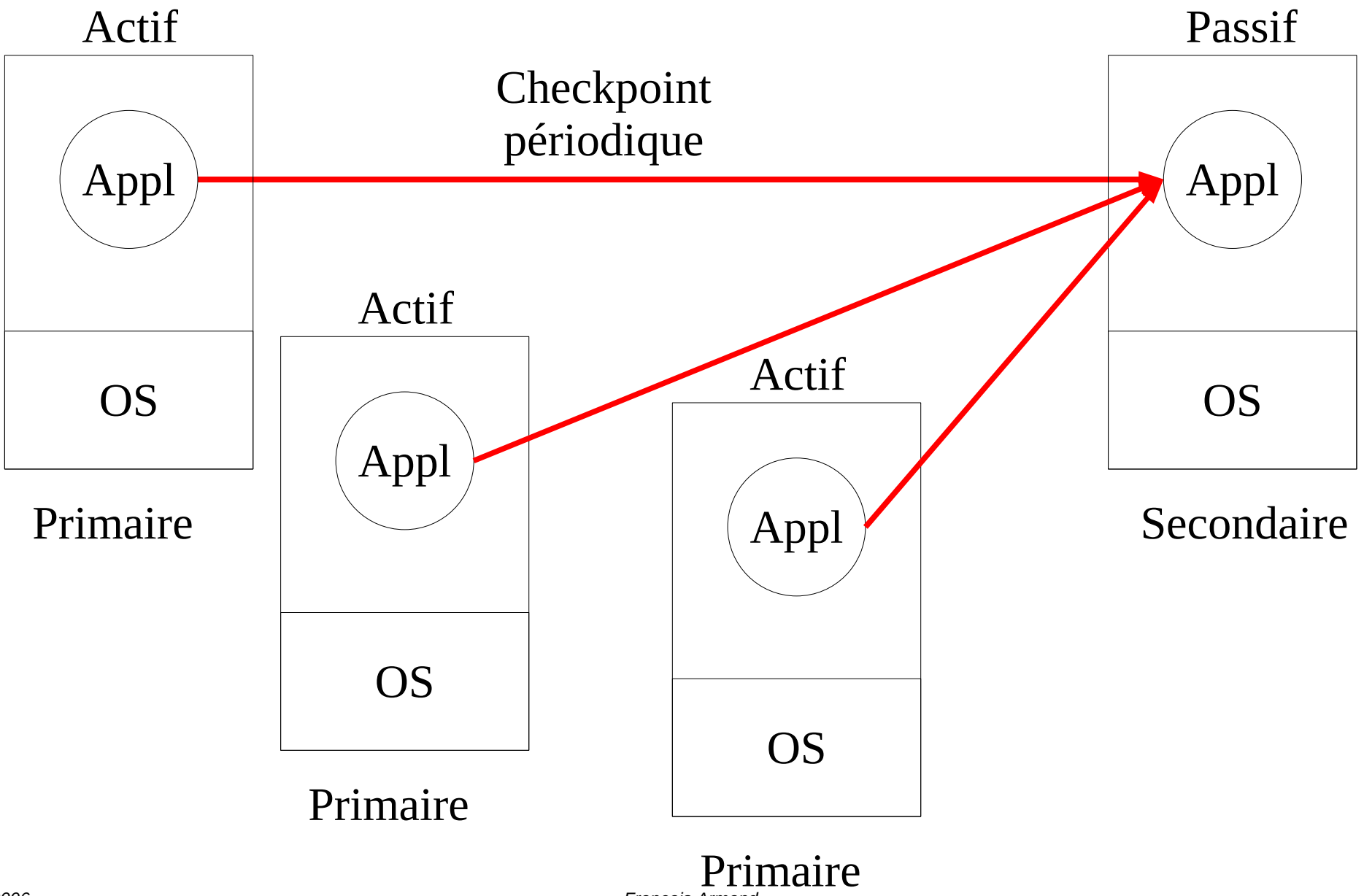




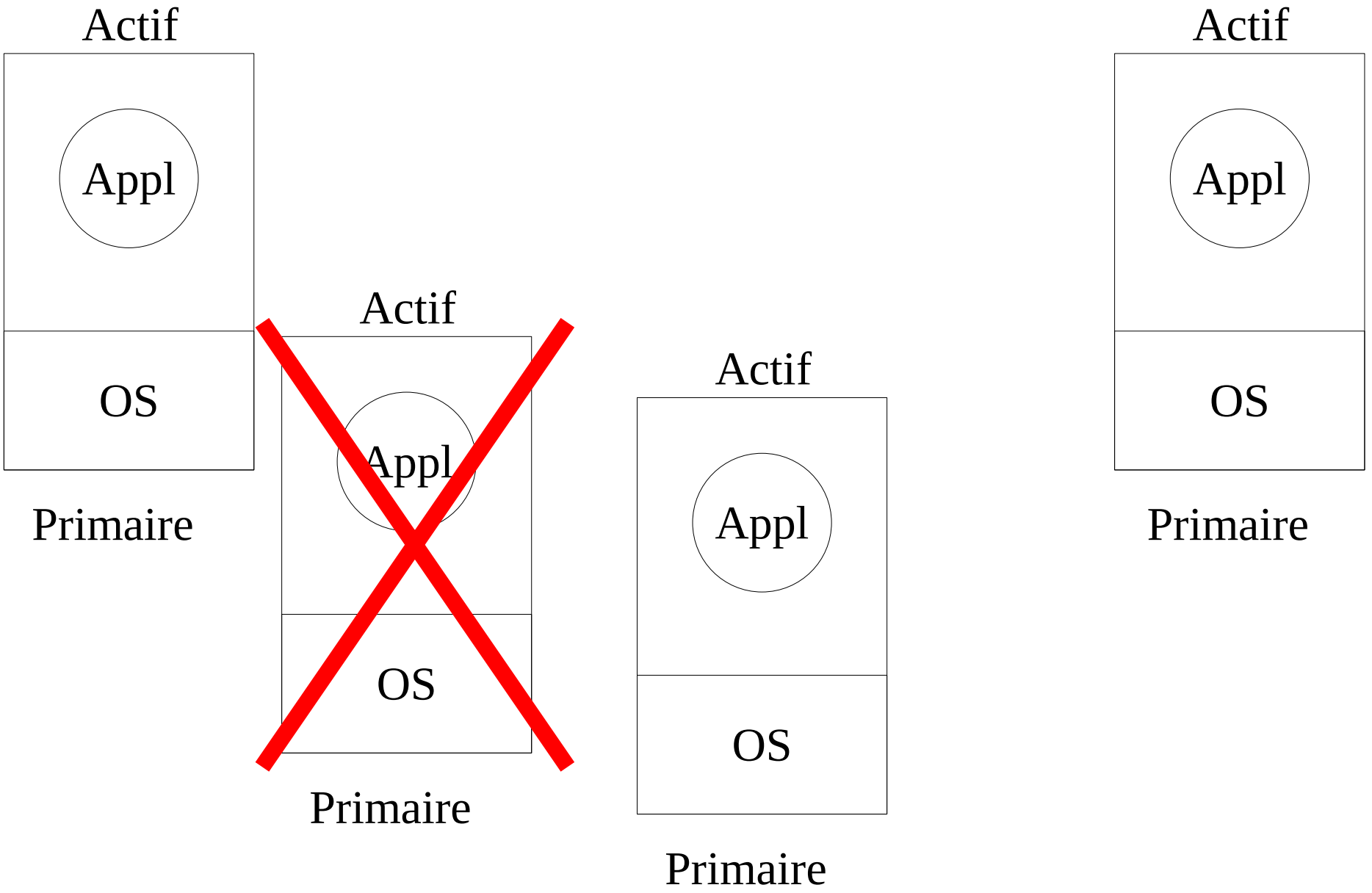
Warm standby: 2N / Load sharing



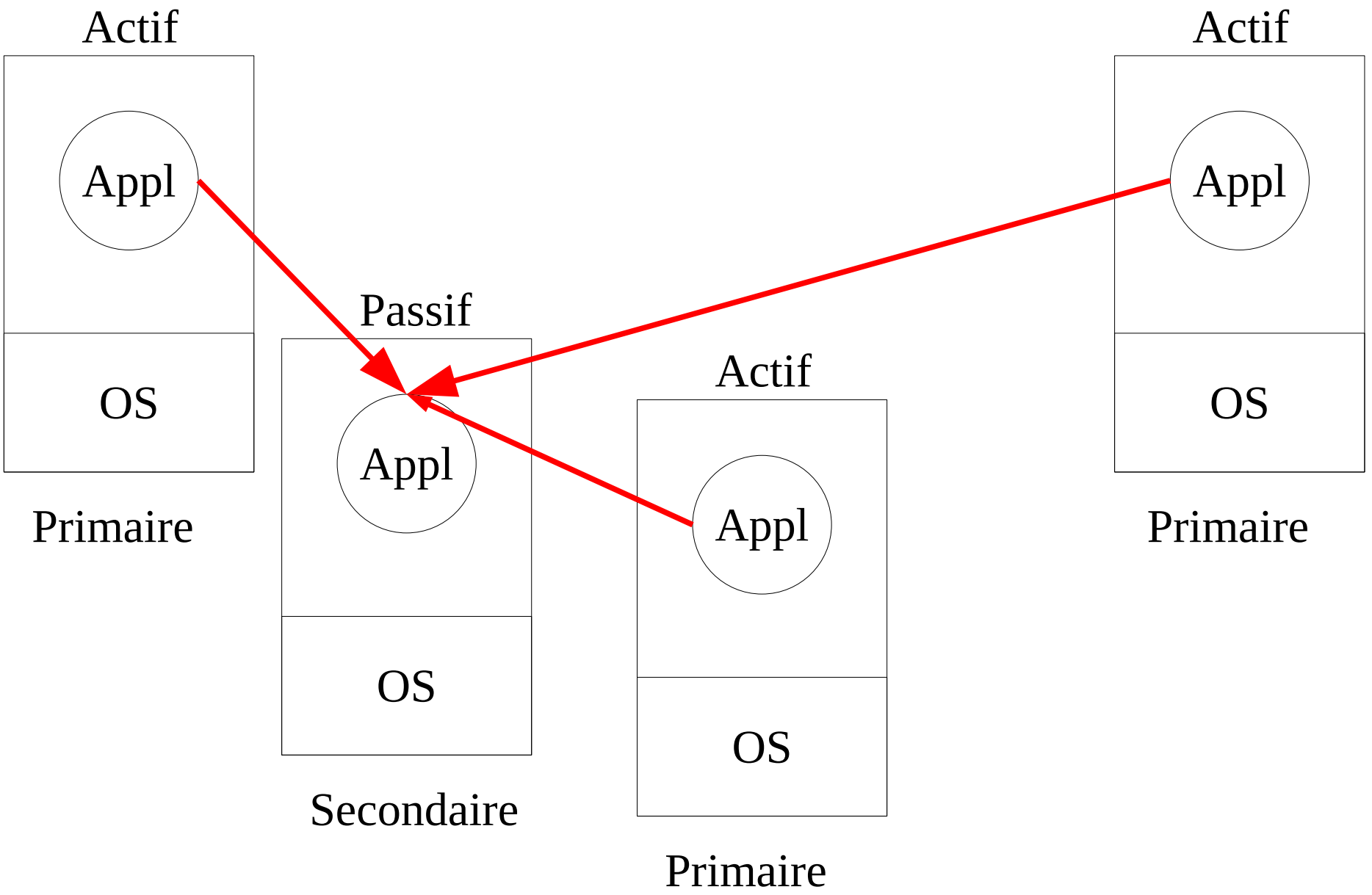
Warm standby: M + 1



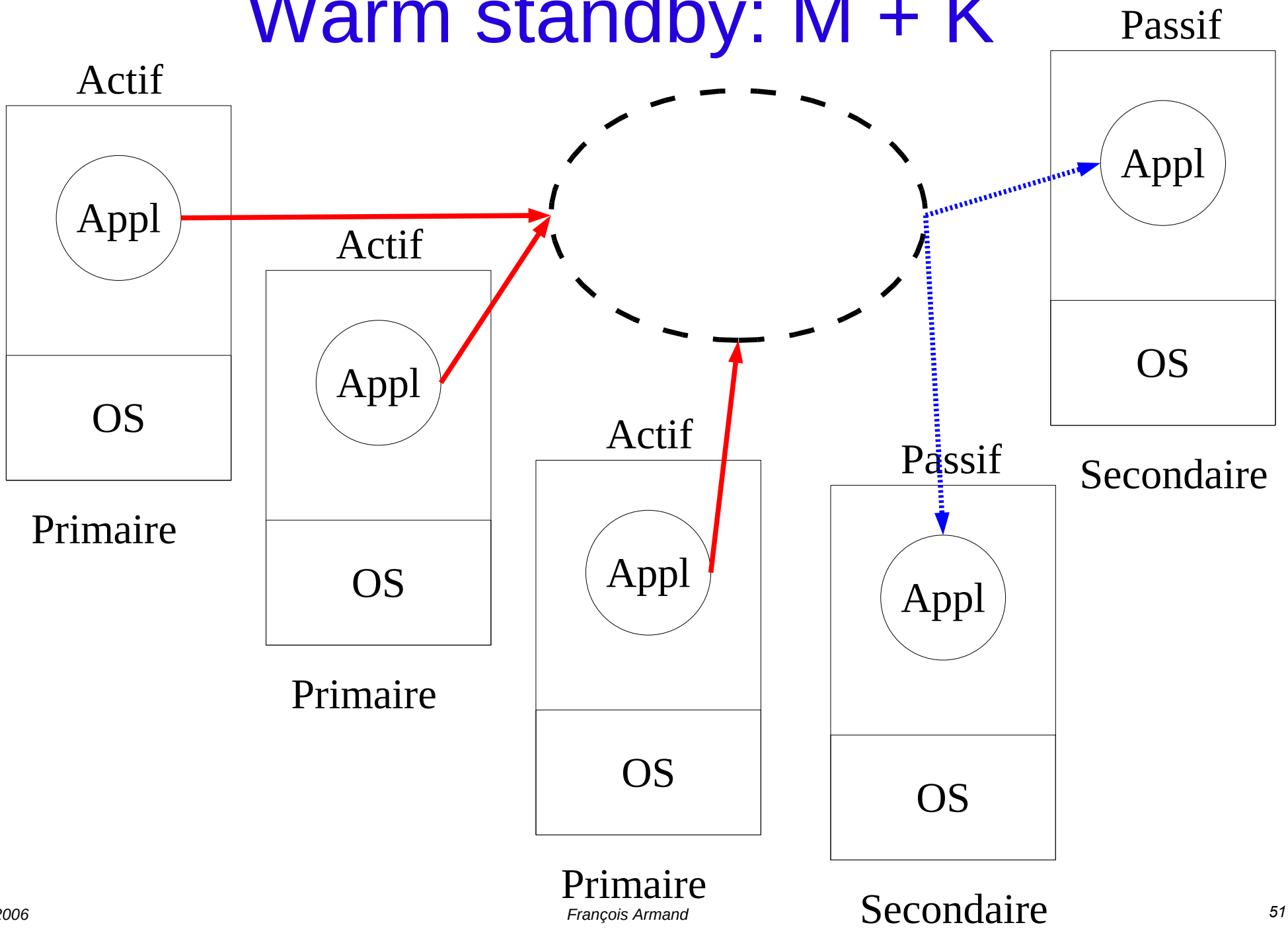
Warm standby: M + 1



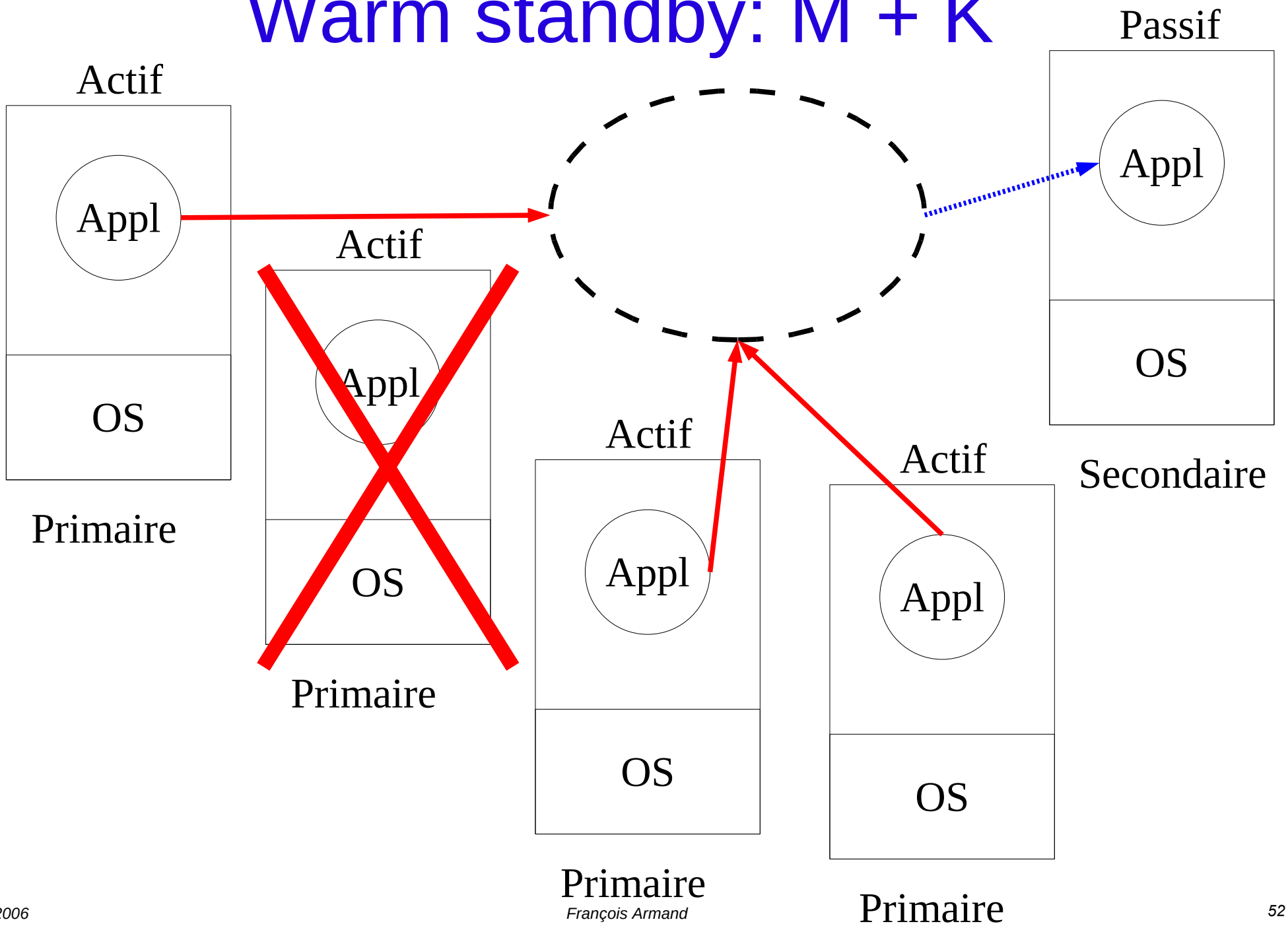
Warm standby: M + 1



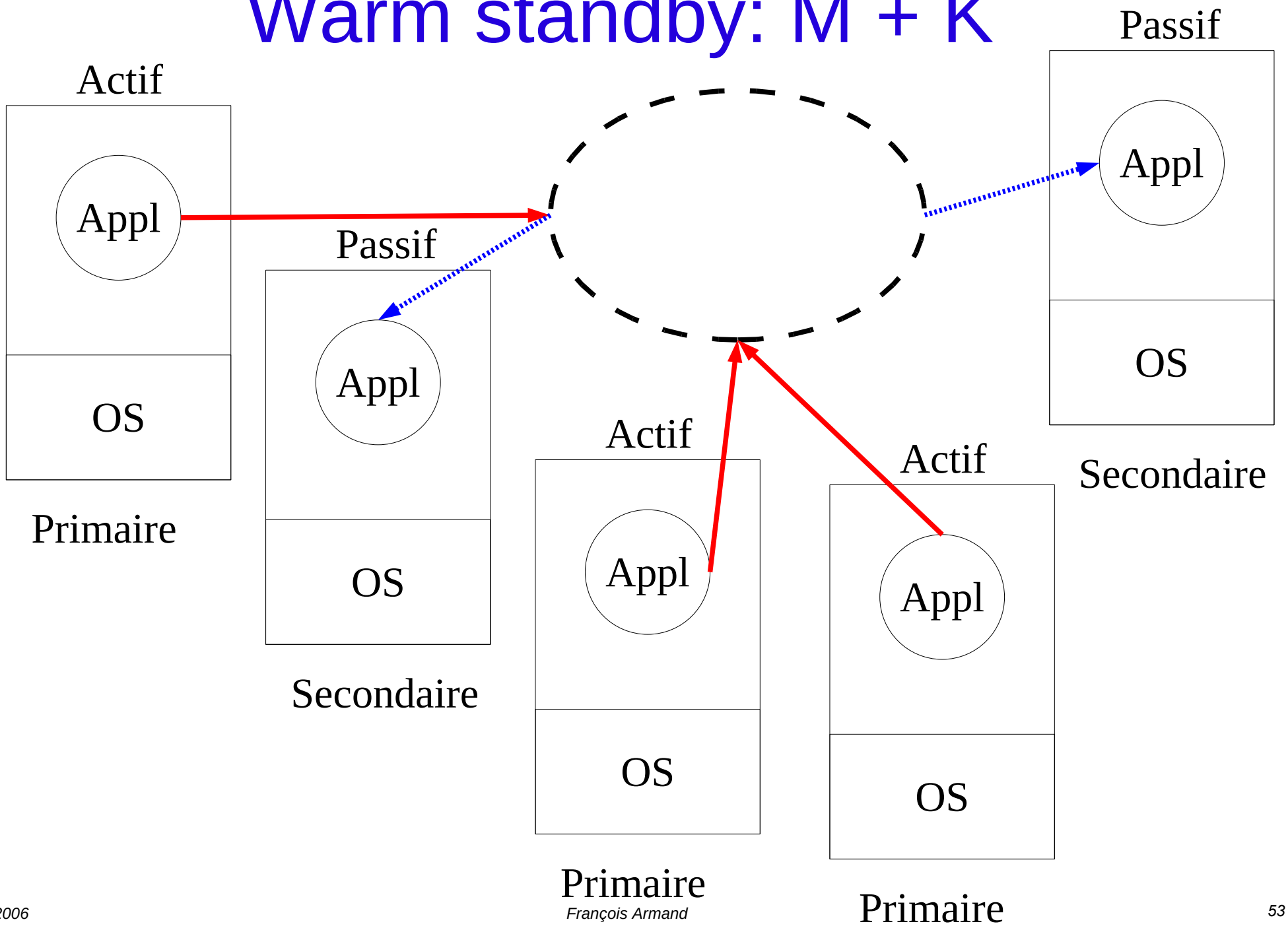
Warm standby: M + K



Warm standby: M + K



Warm standby: M + K



Plan

- Terminologie
- Modélisation
- Ex: Téléphonie
- Moyens locaux à un noeud
- Moyens distribués

Nœud Local

- Améliorer la disponibilité du système en:
- Réduisant les périodes de "downtime"
  - Mécanismes de boot et de démarrage
  - Détection en ligne automatique des périphériques
  - Mécanismes de mise à jour
- Détectant les erreurs au plus tôt:
  - Audits,Diagnostics,
  - Report et enregistrement des erreurs,
  - Watchdog

Nœud Local

- Améliorer la disponibilité du système en:
- Détectant les conditions de surcharge:
  - Instrumentation,
  - Mécanismes d'alerte
- En évitant les fautes:
  - Hardening du système et des drivers
  - Tests, validations, injection de fautes

Nœud Local

- Améliorer la disponibilité du système en:
- Administration en ligne:
  - sauvegarde / restauration,...
- Étant capable de deboguer le système:
  - en ligne: système de traces, boîte noire,...
  - hors ligne: crash dump,
- Gérant les erreurs:
  - Application restart,

Réduire le "downtime"

- Détection insertion / retrait de périphériques:
  - ex: bus cPCI
  - Émission d'une interruption lors de l'insertion d'une carte,
  - Détecter le type de périphérique
  - Charger le pilote approprié,
  - Initialiser le pilote et le périphérique,
  - Signaler l'insertion aux couches applicatives

Réduire le "downtime"

- Mise à jour du système:
  - Nouvelle version de package (rpm),
    - peut se faire en ligne,
    - arrêt / redémarrage des applications,
    - en cas d'erreur: revenir à la version précédente
  - Nouvelle version du système:
    - Nouvelle image système: arrêt / redémarrage
  - Nouvelle installation:
    - Nécessite une partition disponible
    - Transfert des informations de la partition courante vers la nouvelle.

Détecter tôt les erreurs

- Exécuter périodiquement des "audits"
  - Tests logiciels vérifiant le bon état du système, du périphérique,
  - En concurrence avec activité normale du système
  - Fréquence ajustable:
    - En situation de surcharge, cette fréquence sera abaissée
  - Erreurs reportées vers une couche applicative
    - "Fault Manager" qui décide de la conduite à tenir

Détecter tôt les erreurs

- Procéder à des diagnostics
  - Tests d'un périphérique,
  - Nécessite la mise hors service du périphérique:
    - Tests de validation de la carte
  - Service suspendu ou reporté sur un autre périphérique
  - Réalisé en cas de doute, d'erreurs fréquentes
  - Erreurs reportées vers une couche applicative
    - "Fault Manager" qui décide de la conduite à tenir

Reporter les erreurs

- Erreurs détectées par le système:
  - "Remontées" vers la couche applicative
  - Utiliser "syslog" et un filtre...
  - Mécanisme approprié:
    - "system events" (ex: Solaris, C5)
  - Démon utilisateur:
    - Enregistre dans un fichier journal,
    - Transmet aux Fault Manager et aux parties intéressées
    - Transmet à distance (console de management)

Détecter (tôt) les erreurs

- Watchdog à deux phases:
  - Initialisé avec deux temps donnés (A) et (B)
  - Doit être régulièrement réinitialisé avant (A) en général par processus utilisateur
  - Si il n'a pas été réinitialisé dans le délai (A)
    - Génération d'une interruption
    - Si système pas redémarré par software avant délai (B)
    - Sinon => reset hardware
- Pour couvrir initialisation et arrêt:
  - Périodes ajustables de prise en charge par le système

Instrumentation

- Contrôler l'activité du système:
  - Compteur: toujours incrémenté
    - ex: nombre de paquets IP reçus
  - Jauge: valeur oscillant entre un min et un max
    - ex: mémoire physique occupée, utilisation CPU,...
  - Watermark: min / max atteints par une jauge
    - Peuvent être remis à zéro
  - Seuil (threshold): niveau d'alerte associé à une jauge
    - Ajustable par administration, génère un événement,
    - Plusieurs seuils par jauge

Hardening système

- Éliminer (si nécessaire) les panics inutiles
  - Lazy panic
- Pilotes:
  - Fautes liées à un périphérique ne doivent pas se propager en dehors du pilote
  - Notifier les erreurs détectées
  - Se protéger contre les données corrompues
  - Se protéger contre les interruptions "permanentes"

Administration en ligne

- Systèmes de Fichiers:
  - Sauvegarde et restauration d'images cohérentes sans arrêter le service,
- Ajout / retrait dynamique de disques:
  - Logical Volume Management (LVM, EVMS, VxVM)
  - Agrandir / réduire la taille des partitions en services,
  - Modifier dynamiquement le mirroring,
  - Agrandir / réduire la taille des systèmes de fichiers en service,

Deboguer en Opérations

- Impossibilité de changer le système avec version de debug,
- Impossibilité de suspendre le service,
- Traces conditionnelles dynamiques,
  - Activables par commande administration,
  - Modulaires: par zone fonctionnelle, par niveau
  - Stockées dans un fichier,...
- Concept de boîte noire
  - Buffer circulaire toujours partie du "crash dump"
  - Dernières xx (30) secondes du système

Crash Dump

- Vidage mémoire en cas de panic
  - Sauvegarder sans écraser au redémarrage
  - Gérer plusieurs fichiers crash dump
    - En cas de crash récurrents, le premier dump est plus utile
  - Support de différents médias:
    - disque, réseau, ligne série,...
    - Vitesses différentes,
  - Configurer le contenu
    - Partiel sur media lent,
    - Complet sur media rapide ou en cas de besoin "vital"

Redémarrage d'applications

- Redémarrer à froid
  - Ressource group de Sun Cluster, Wolf,...
  - Keepalive (Linux)
- Redémarrer avec un état
  - Checkpoint:
    - Nombreuses tentatives: http://www.checkpointing.org
    - Délicat pour les connexions réseau sauf:
    - Rocks: http://www.cs.wisc.edu/~zandy/rocks

Plan

- Terminologie
- Modélisation
- Ex: Téléphonie
- Moyens locaux à un noeud
- Moyens distribués

Détection

- Mécanismes de heart-beat, Failure detectors
- Problèmes de consensus
  - Qui est vivant, qui est hors-service?
- Ex: Linux-HA
  - STONITH

Mise à jour système

- Rolling upgrade:
  - Mettre à jour un noeud après l'autre
  - L'activité du noeud mis à jour est assurée temporairement par son backup
  - S'assurer que la mise à jour converge
- Split upgrade:
  - On arrête la moitié du système et on le met à jour
  - On bascule l'activité de la vieille moitié vers la nouvelle
  - On arrête et on redémarre la vieille moitié.