

# Video du banc de Tests COMON

COMON



Grenoble - Alpes Métropole  
Communauté d'agglomération



Avril 2012

# Outline

- 1 L'outil de test Lurette
- 2 Tests dans COMON - Chaîne sans rupture
- 3 Demo
- 4 Conclusion

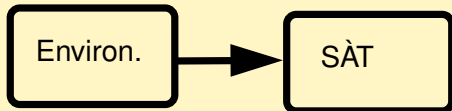
# Plan

- 1 **L'outil de test Lurette**
- 2 Tests dans COMON - Chaîne sans rupture
- 3 Demo
- 4 Conclusion

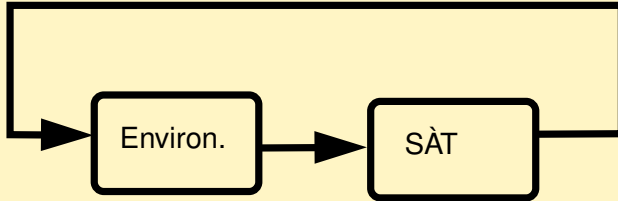
# Lurette - Tests fonctionnels automatisés de systèmes réactifs

- Test **fonctionnel** (boîte noire)
  - ▶ confronter une implémentation et une spécification
- Test **Automatisé**
  - ▶ génération des stimuli (entrées du SàT)
  - ▶ dépouillement du résultat des tests (oracle)
- Basé sur une description **formelle**
  - ▶ des attendus du *Système à Tester* (SàT)
  - ▶ des hypothèses faites sur son environnement
- Systèmes **réactifs**
  - ▶ le SàT réagit à l'environnement qu'il cherche à contrôler (feedback)
  - ▶ un environnement *réaliste* doit faire de même

# Test de systèmes bouclés

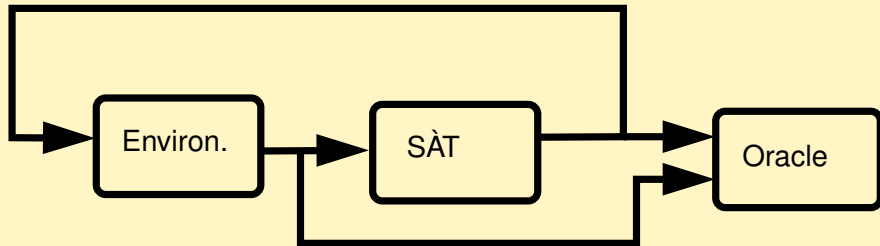


# Test de systèmes bouclés



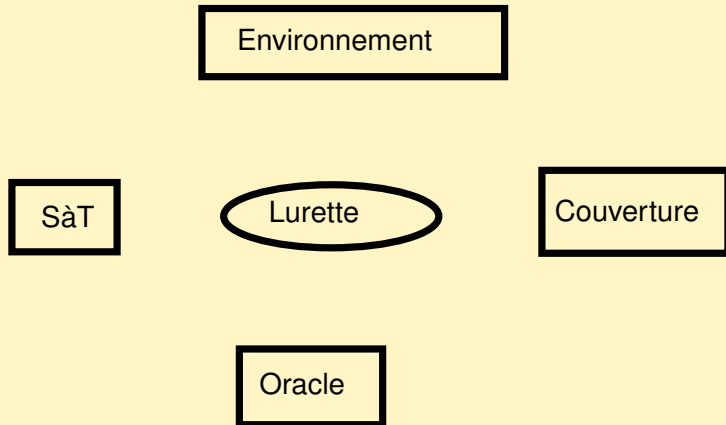
- L'environnement (**Stimulateur**) peut-être vu comme un système réactif non-déterministe.
- Générateur de vecteurs de test sous contrainte

# Test de systèmes bouclés



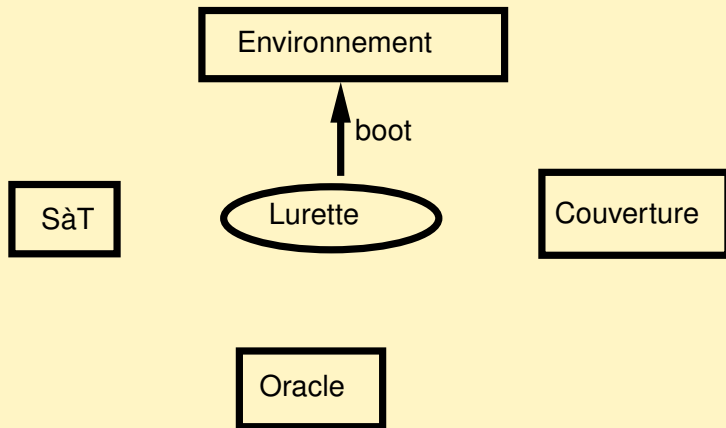
- L'environnement (**Stimulateur**) peut-être vu comme un système réactif non-déterministe.
- Générateur de vecteurs de test sous contrainte

# Flot de données - à chaque cycle

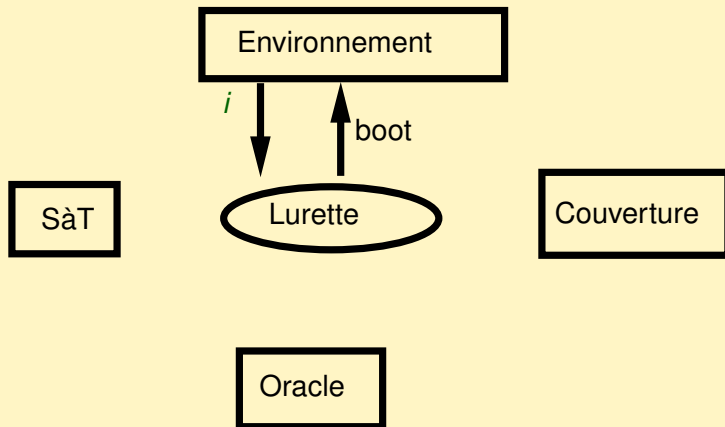




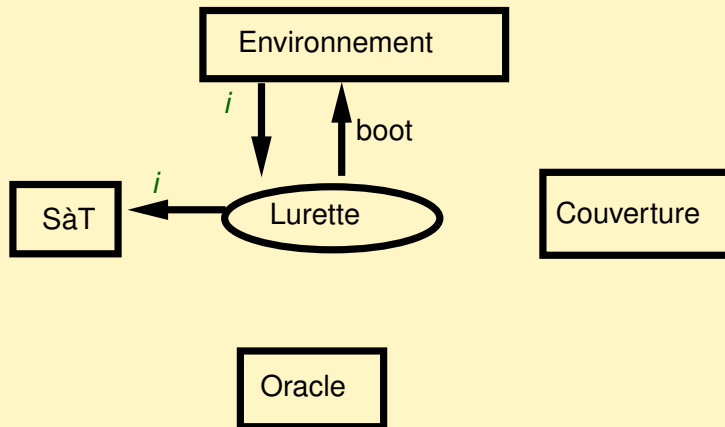
# Flot de données - à chaque cycle



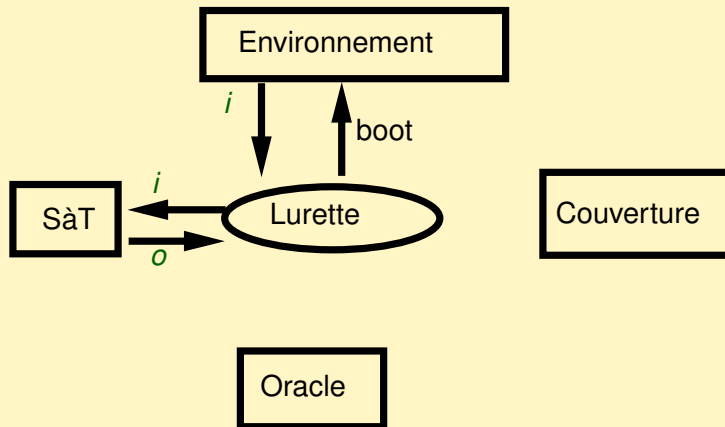
# Flot de données - à chaque cycle



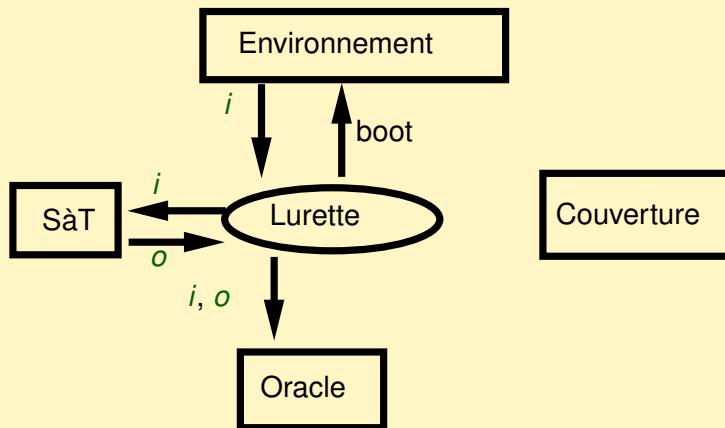
# Flot de données - à chaque cycle



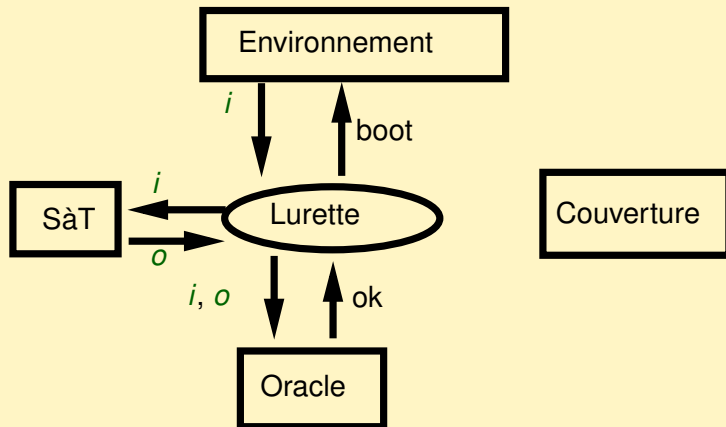
# Flot de données - à chaque cycle



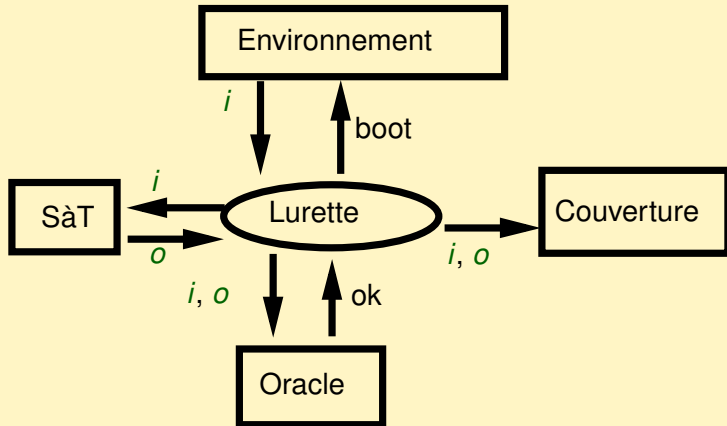
# Flot de données - à chaque cycle



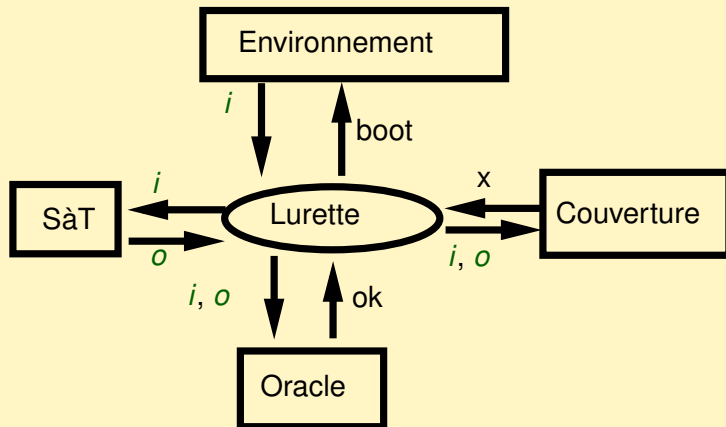
# Flot de données - à chaque cycle



# Flot de données - à chaque cycle

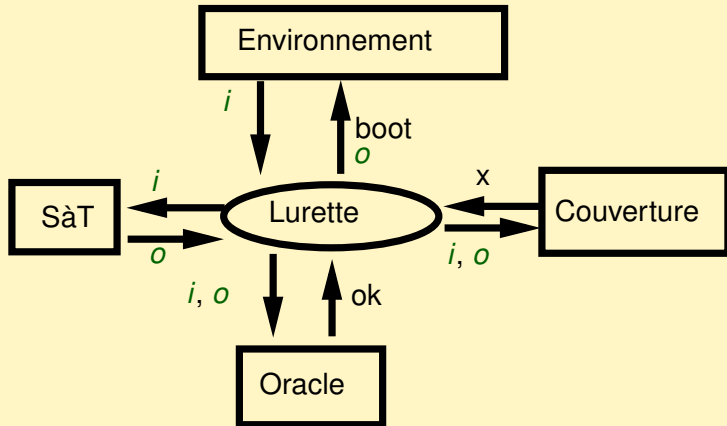


# Flot de données - à chaque cycle

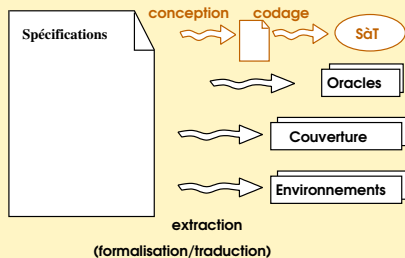




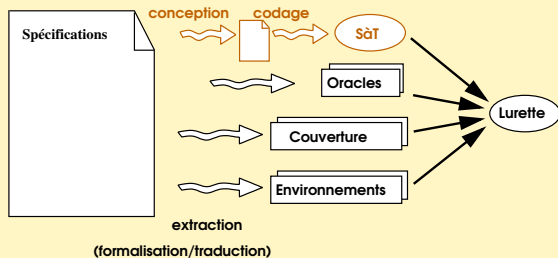
# Flot de données - à chaque cycle



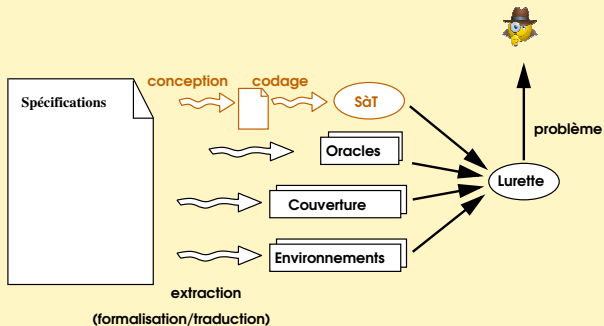
# Processus de test Lurette



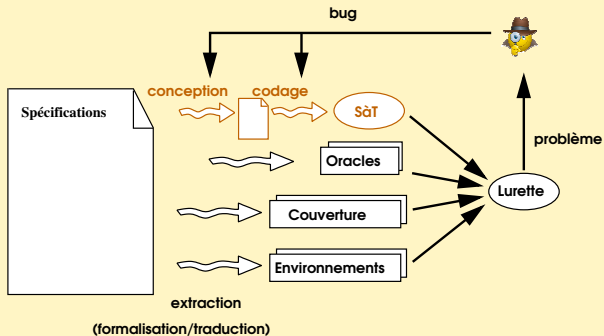
# Processus de test Lurette



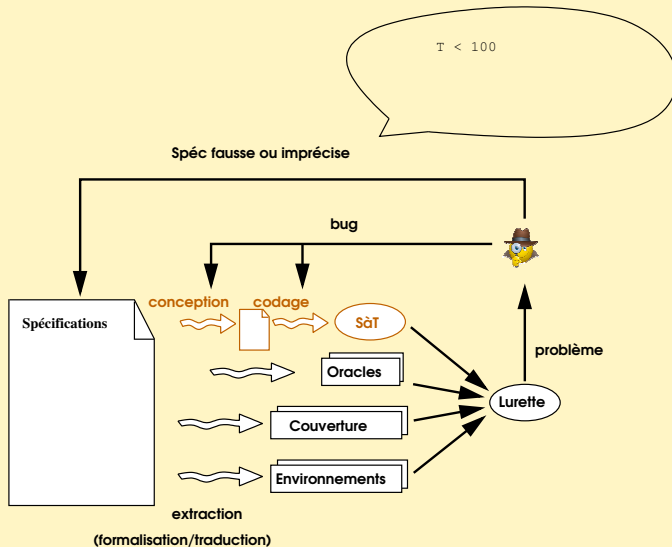
# Processus de test Lurette



# Processus de test Lurette



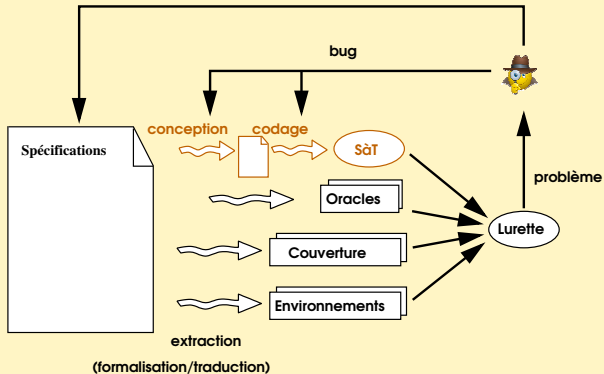
# Processus de test Lurette



# Processus de test Lurette

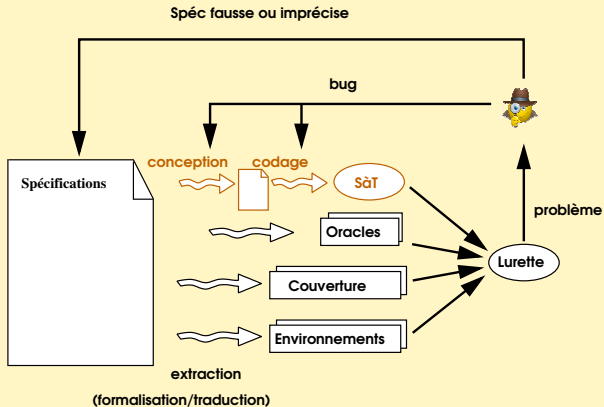
$T < 100$   
valid =>  $T < 100$

Spéc fausse ou imprécise



# Processus de test Lurette

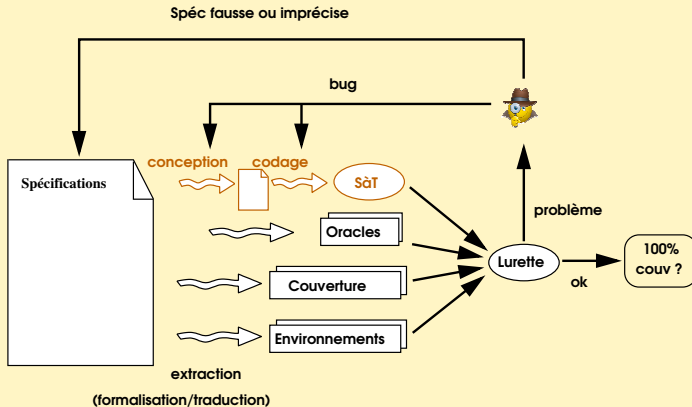
~~$T < 100$~~   
~~valid  $\Rightarrow T < 100$~~   
 valid and nominal  $\Rightarrow T < 100$





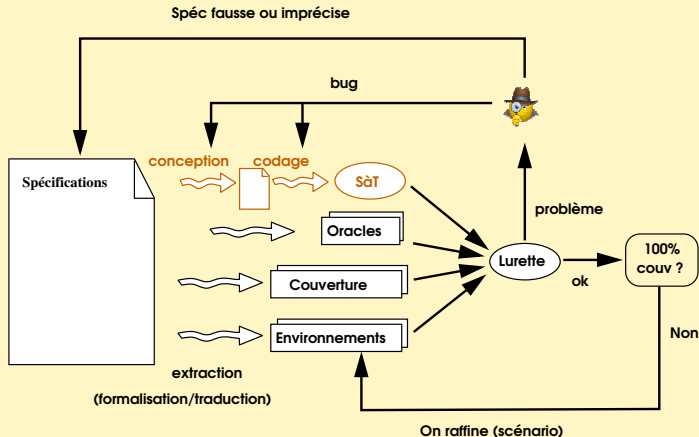
# Processus de test Lurette

~~$T < 100$~~   
~~valid  $\Rightarrow T < 100$~~   
 valid and nominal  $\Rightarrow T < 100$



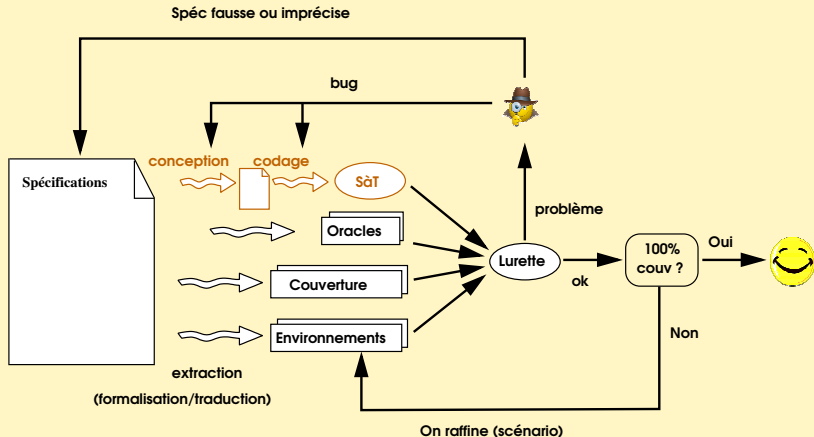
# Processus de test Lurette

~~$T < 100$~~   
~~valid  $\Rightarrow T < 100$~~   
 valid and nominal  $\Rightarrow T < 100$



# Processus de test Lurette

~~$T < 100$~~   
~~valid  $\Rightarrow T < 100$~~   
 valid and nominal  $\Rightarrow T < 100$



# Plan

1 L'outil de test Lurette

**2 Tests dans COMON - Chaîne sans rupture**

3 Demo

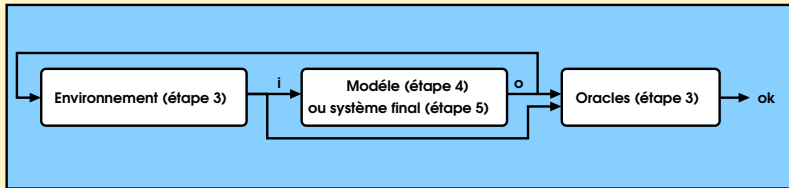
4 Conclusion

# Test et Chaîne sans rupture COMON

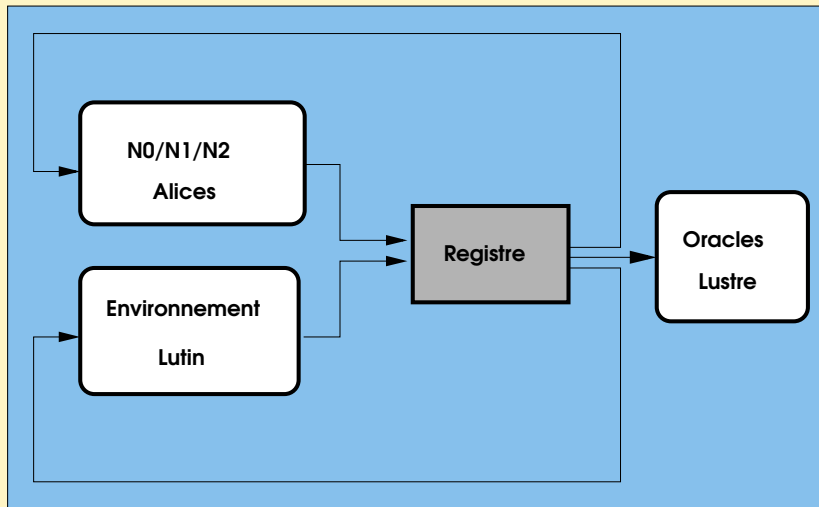
1. Analyse des besoins
2. Spécification des exigences fonctionnelles en langue naturelle et à l'aide de dessins informels
3. Formalisation de ces exigences fonctionnelles (Oracles)
4. Obtention au plus tôt d'un modèle abstrait et exécutable satisfaisant ces exigences (Alices)
5. Obtention d'une implémentation finale satisfaisant ces exigences

# Test et Chaîne sans rupture COMON

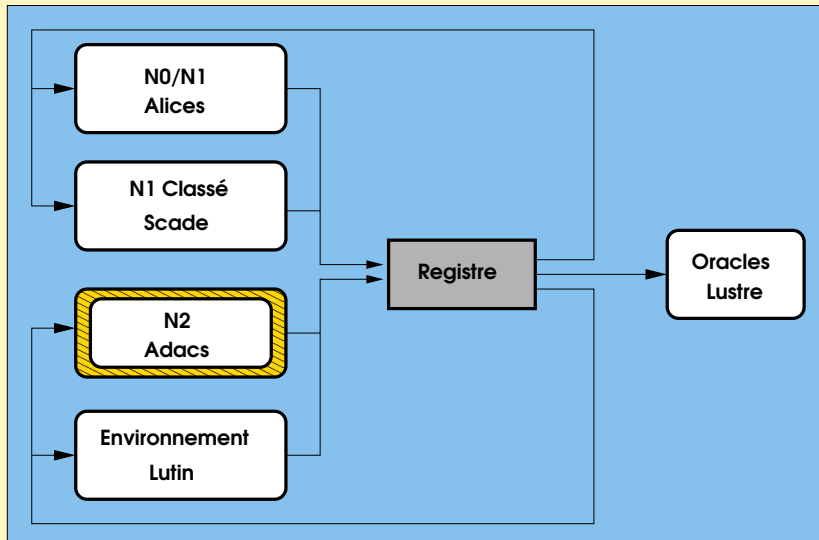
1. Analyse des besoins
2. Spécification des exigences fonctionnelles en langue naturelle et à l'aide de dessins informels
3. Formalisation de ces exigences fonctionnelles (Oracles)
4. Obtention au plus tôt d'un modèle abstrait et exécutable satisfaisant ces exigences (Alices)
5. Obtention d'une implémentation finale satisfaisant ces exigences



# Banc de test en mode simulation

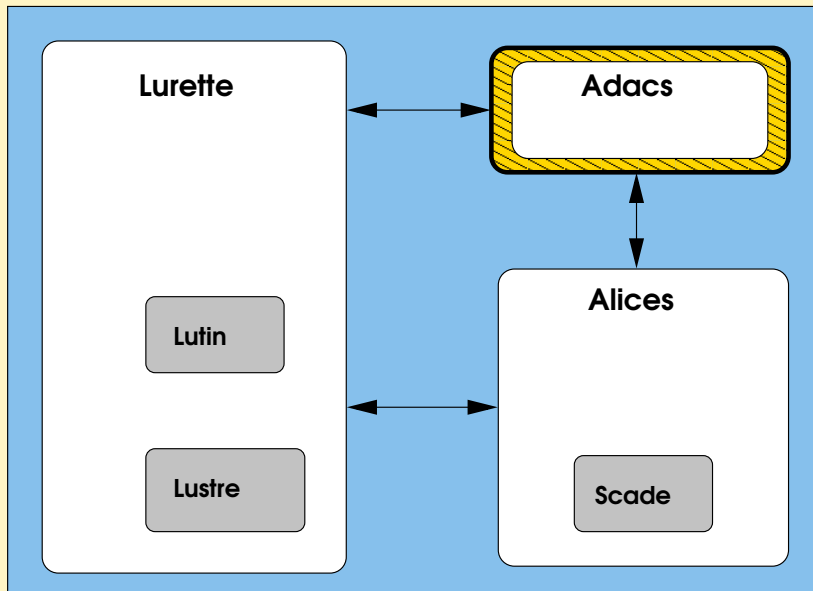


# Banc de test en mode stimulation

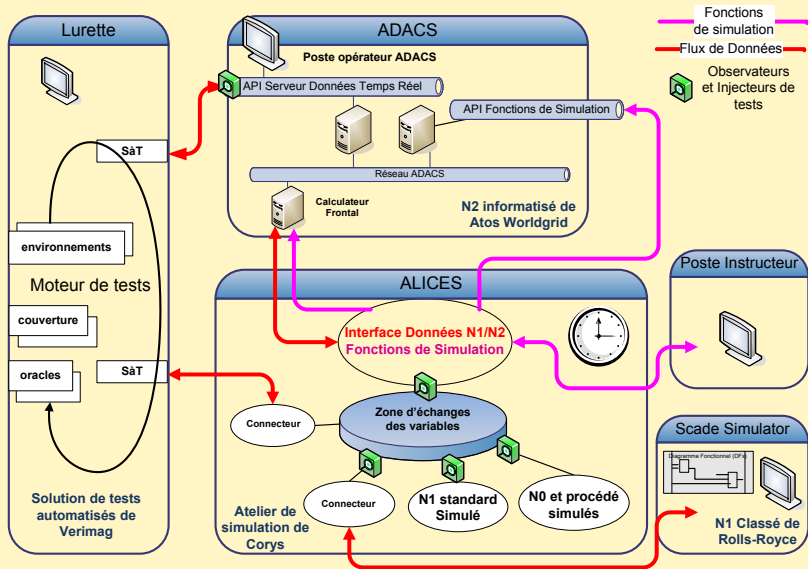




# Architecture logicielle du banc test



# Architecture logicielle du banc test - vue détaillée

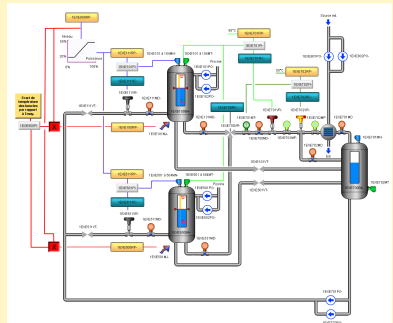


# Plan

- 1 L'outil de test Lurette
- 2 Tests dans COMON - Chaîne sans rupture
- 3 Demo**
- 4 Conclusion

# Le cas d'étude COMON

- Une étude de cas représentative
  - ▶ D'un circuit hydraulique nucléaire
    - Circuit physique
    - Capteurs et actionneurs
  - ▶ Et de son contrôle-commande
    - Redondance
    - Régulation
    - Sûreté
  - ▶ Son démonstrateur
    - Procédé simulé
    - N1 standard simulé
    - N1 classé émulé
    - N2 réel



# Ce que l'on voit dans la Démo

- 4 ateliers hétérogènes communiquer
- Scénarios de stimulation (Environnement)
  - ▶  $n$  (0, puis 1, puis 2) pannes au hasard, avec comme contrainte d'éviter (si possible) l'action de sûreté
  - ▶ Un opérateur virtuel bouclé
    - Une consigne cible est choisie au hasard
    - L'opérateur virtuel change la consigne d'au plus « pas » pour se rapprocher de la cible ; il attend la stabilité du sàt avant de la changer la consigne à nouveau (boucle 1)
    - quand la consigne atteint sa cible, on rechoisit une cible (boucle 2)
- Propriétés Invalidées (Oracles)
  - ▶ Transitions entre états du système (nominal, 2/3-1/3, dégradé, urgence) : anomalie dans le calcul de la situation sur le N2
  - ▶ Seuil haut dépassé : anomalie dans le voteur N1 qui n'élimine pas le capteur en panne d'une fausse moyenne



# Plan

- 1 L'outil de test Lurette
- 2 Tests dans COMON - Chaîne sans rupture
- 3 Demo
- 4 Conclusion**

# Conclusion

- 4 ateliers hétérogènes communiquent
- L'approche « orientée par les modèles »
  - ▶ Simulation et validation précoce (à toutes les étapes)
  - ▶ Développement incrémental/chaîne sans rupture
- Une mise en œuvre sur un cas d'étude simple en simulation puis en stimulation
  - ▶ de la formalisation de propriétés fonctionnelles du système
  - ▶ de la formalisation d'exigences de couverture fonctionnelle
  - ▶ de la formalisation de propriétés des environnements
  - ▶ la génération automatique des vecteurs de tests et vérification automatique des attendus