

# Front-End pour compilateur dataflow utilisant l'infrastructure llvm

*Semba 2013*

Mickaël Dardaillon Kevin Marquet Tanguy Risset  
Jérôme Martin Henri-Pierre Charles



Rhône-Alpes <sup>Région</sup>

*inria* informatics mathematics



# Plan

- 1 Problématique
  - Application 3GPP/LTE
  - Modèle de calcul
- 2 Chaîne de compilation
  - Infrastructure llvm
  - Front-End
- 3 Plateforme Magali
- 4 Conclusion

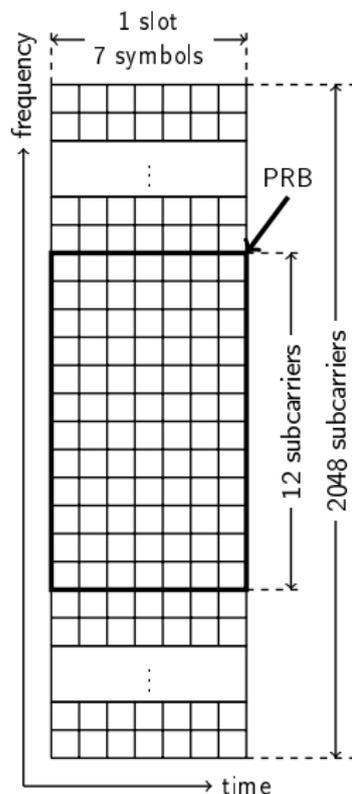
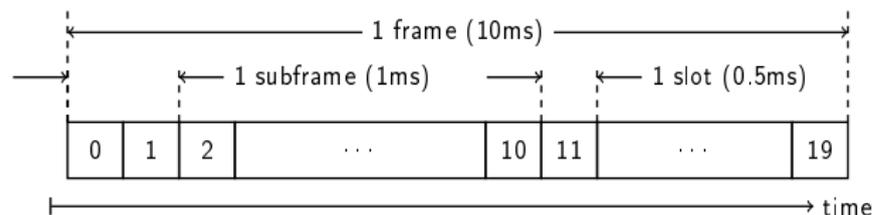
# Application visée : 3GPP/LTE

Release 8, mode 5 : réception

## Caractéristiques :

- Bande passante de 20 MHz
- MIMO 4x2
- Débit entrant : 2 Gb/s
- Consommation : 500 mW

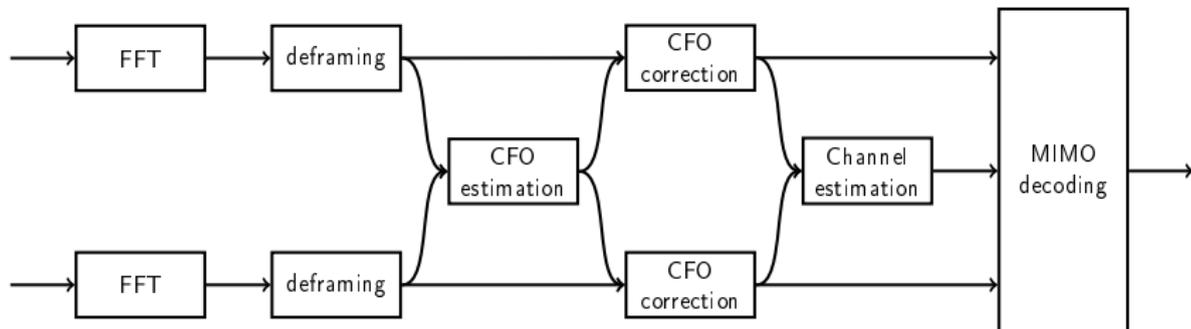
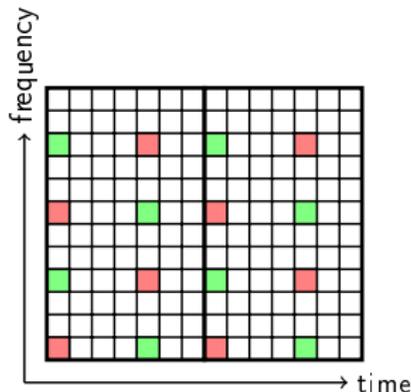
## Traitements décomposable en 2 parties



# Traitements

## décodage OFDMA

- statique
- gourmande en ressources



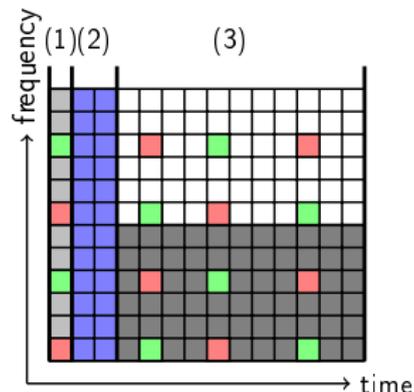
# Traitements

## décodage OFDMA

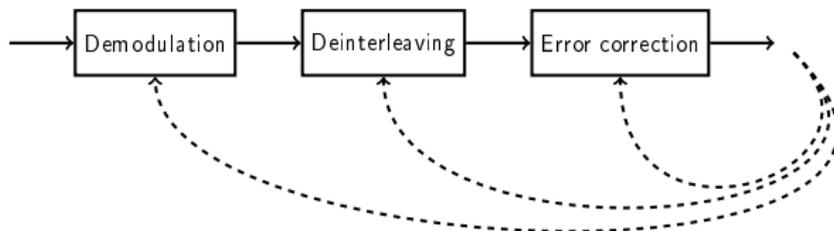
- statique
- gourmande en ressources

## décodage canal

- dynamique



- (1) Format (PCFICH)  
 (2) Control (PDCCH)  
 (3) Data (PDSCH)



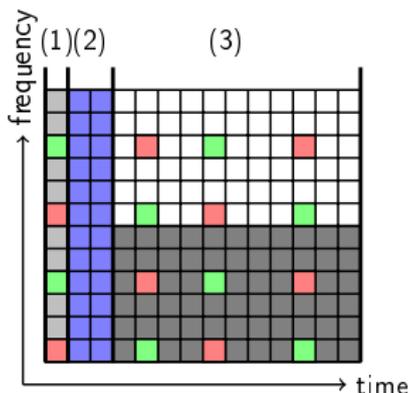
# Traitements

## décodage OFDMA

- statique
- gourmande en ressources

## décodage canal

- dynamique

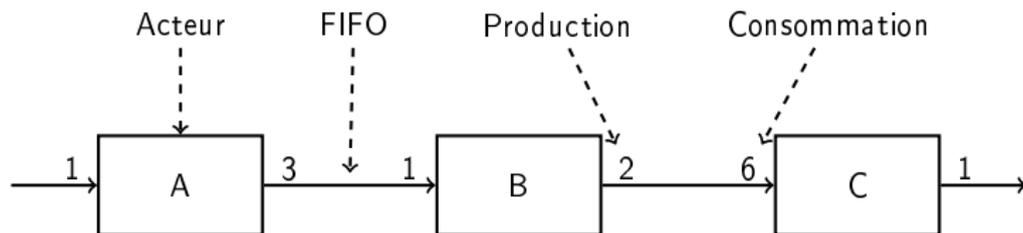


## Problématique :

*Quel modèle de calcul pour représenter l'application 3GPP/LTE ?*

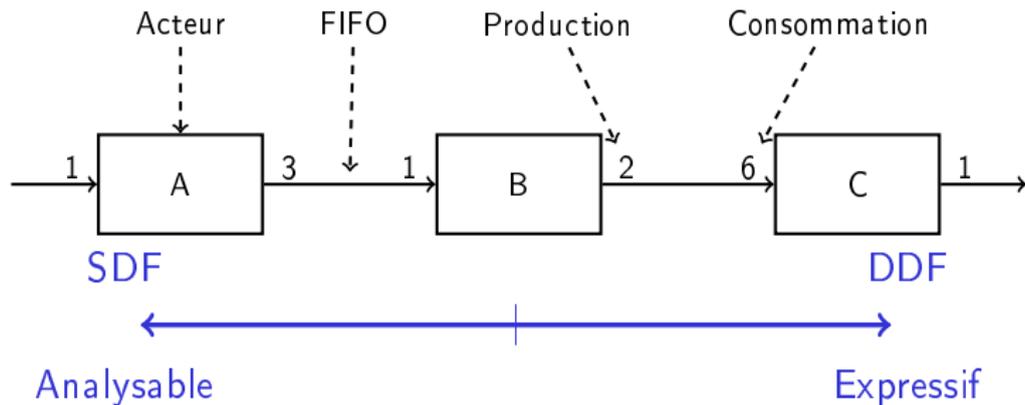
# Modèle de Calcul

## DataFlow



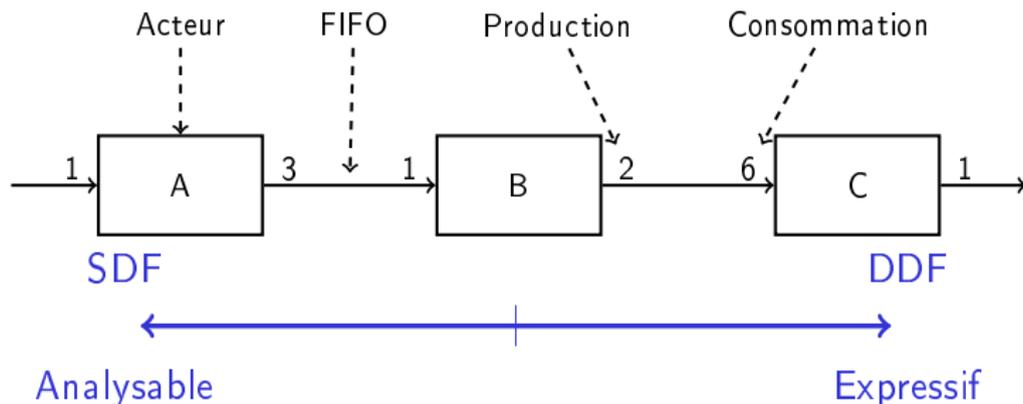
# Modèle de Calcul

## DataFlow



# Modèle de Calcul

## DataFlow



### Statique :

- StreamIt, MIT
- ΣC, CEA LIST

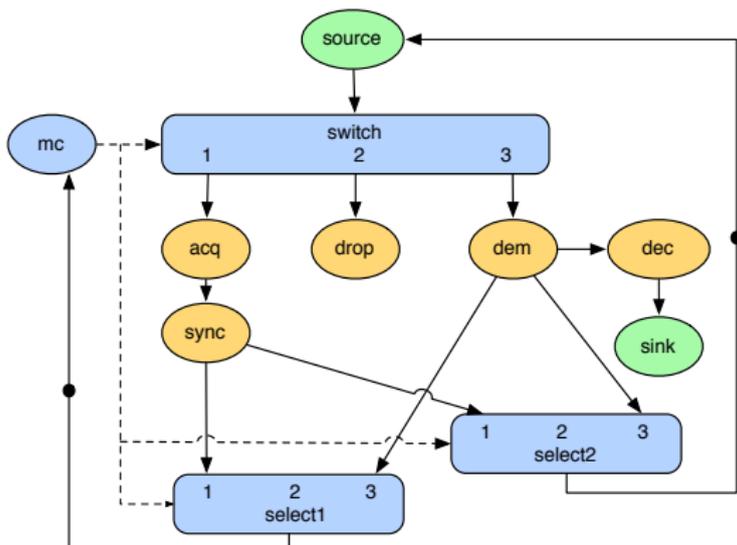
### Dynamique :

- MAPS, RWHT Aachen University
- SPEX, University of Michigan

# Modèle de calcul (suite)

## MCDF, ST-Ericsson Eindhoven

- Analysable
- Ordre quasi-statique



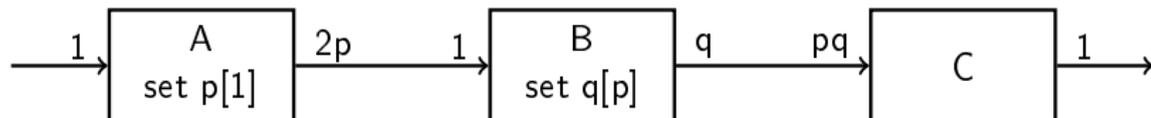
# Modèle de calcul (suite)

## MCDF, ST-Ericsson Eindhoven

- Analysable
- Ordre quasi-statique

## SPDF, Inria Grenoble

- Analysable
- Ordonnancement quasi-statique



## Modèle de calcul (suite)

### MCDF, ST-Ericsson Eindhoven

- Analysable
- Ordre quasi-statique

### SPDF, Inria Grenoble

- Analysable
- Ordonnancement quasi-statique

### Problématique :

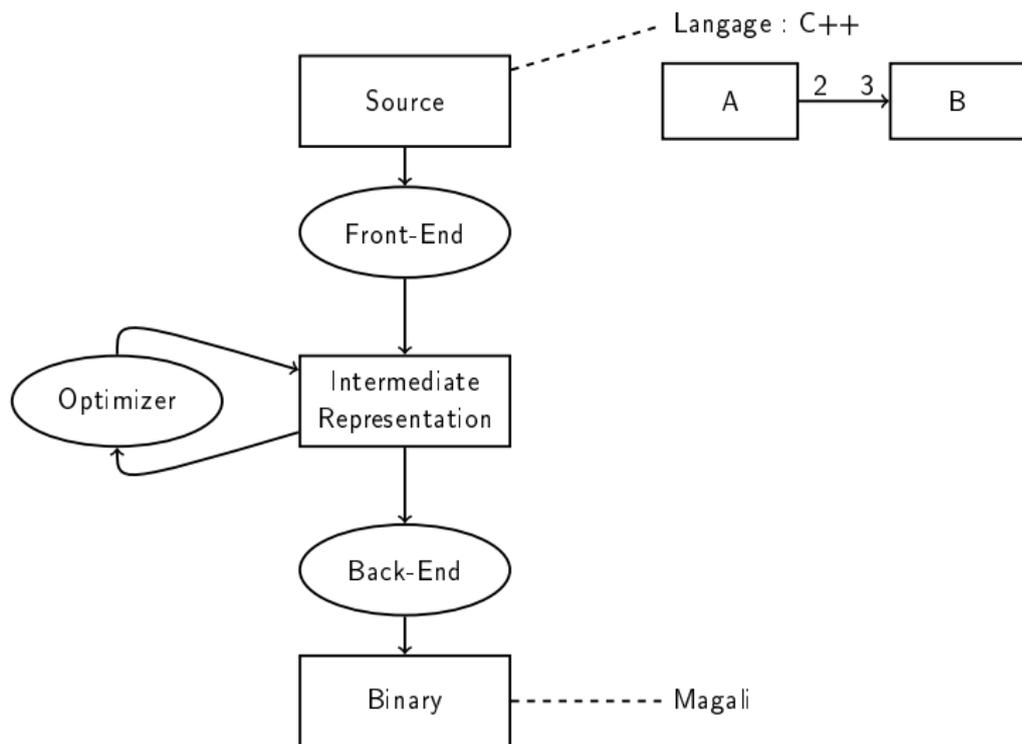
*Quel chaîne de compilation pour implémenter ce modèle de calcul ?*

# Plan

- 1 Problématique
  - Application 3GPP/LTE
  - Modèle de calcul
- 2 Chaîne de compilation
  - Infrastructure llvm
  - Front-End
- 3 Plateforme Magali
- 4 Conclusion

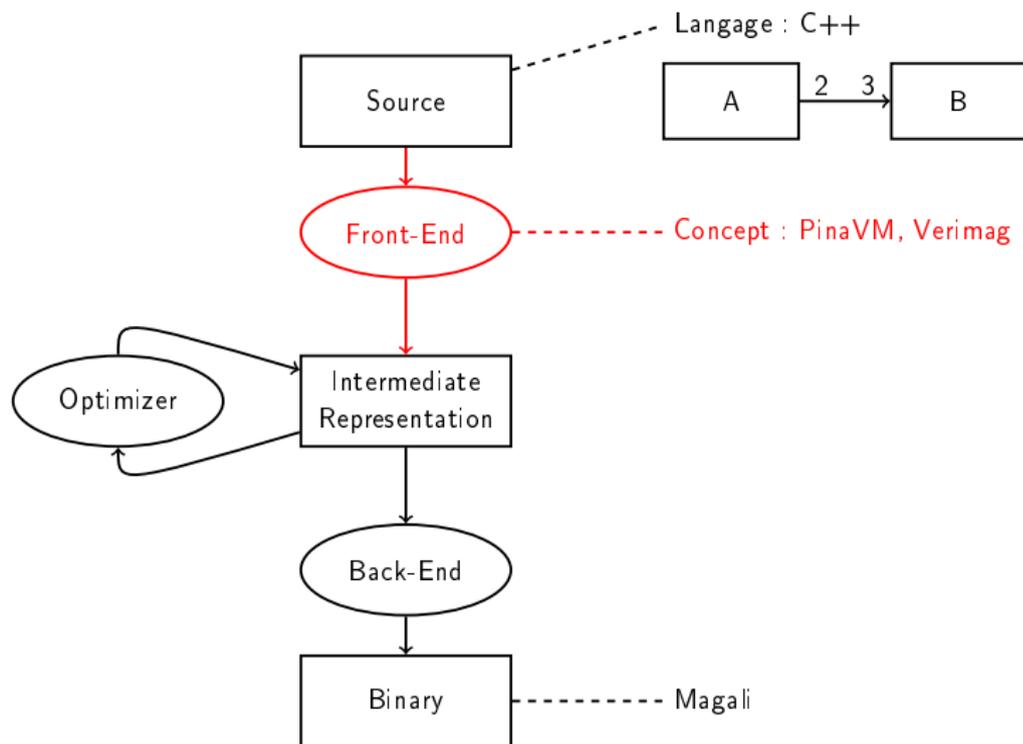
# Chaîne de compilation

Objectif :



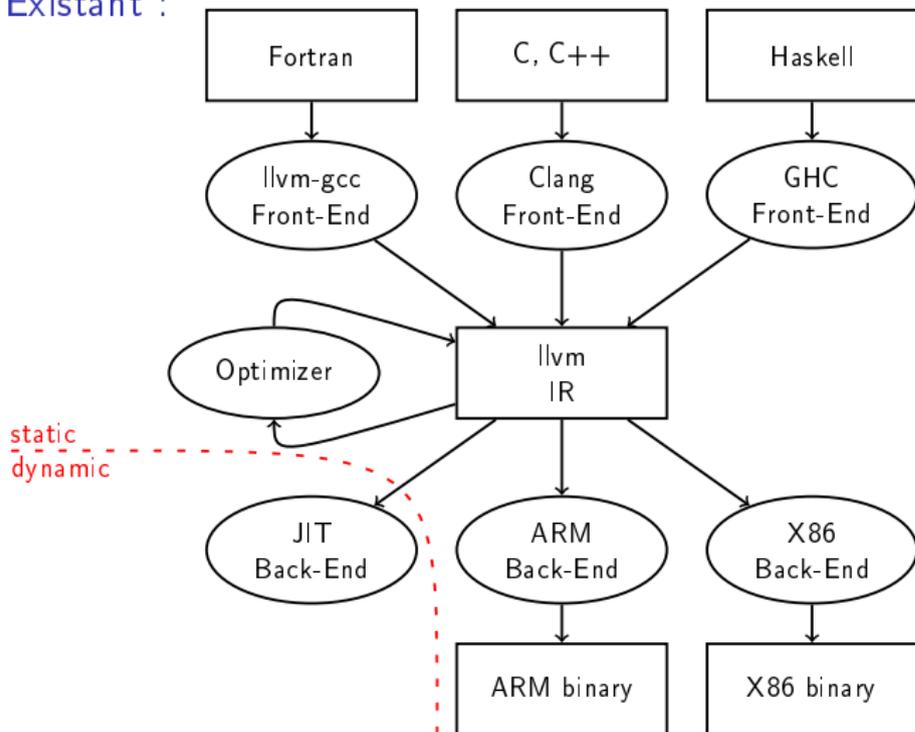
## Chaîne de compilation

Objectif :



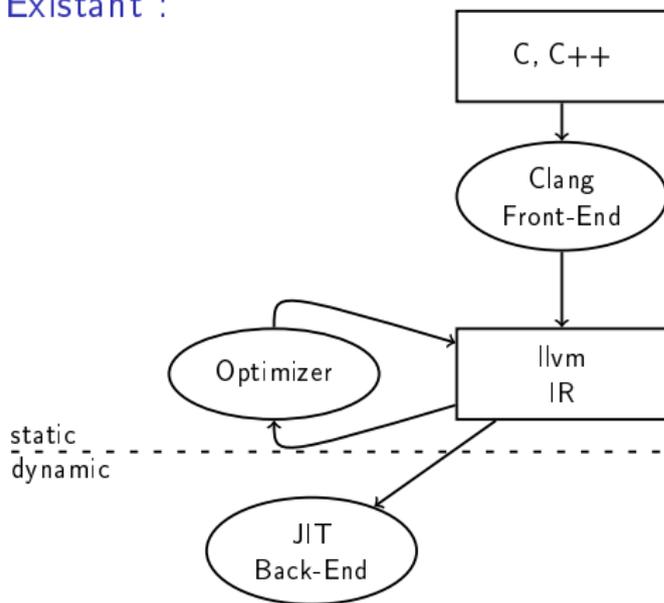
# Infrastructure llvm

Existant :



# Front-End

Existant :

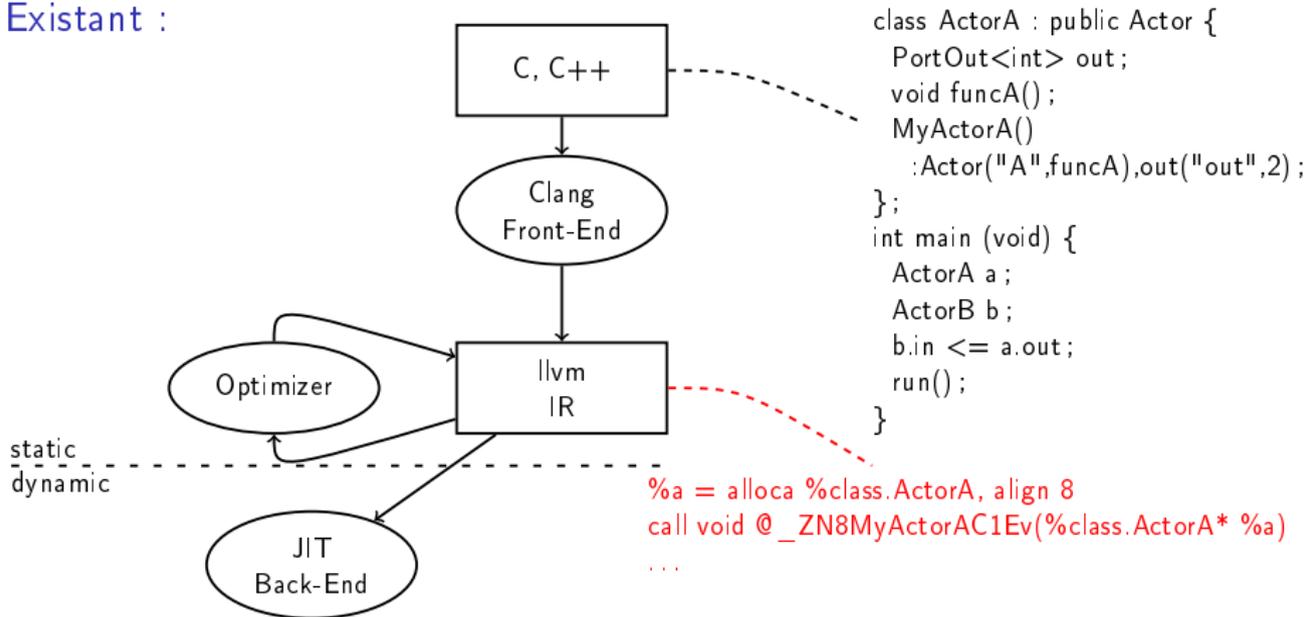


```

class ActorA : public Actor {
    PortOut<int> out;
    void funcA();
    MyActorA()
        :Actor("A",funcA),out("out",2);
};
int main (void) {
    ActorA a;
    ActorB b;
    b.in <= a.out;
    run();
}
  
```

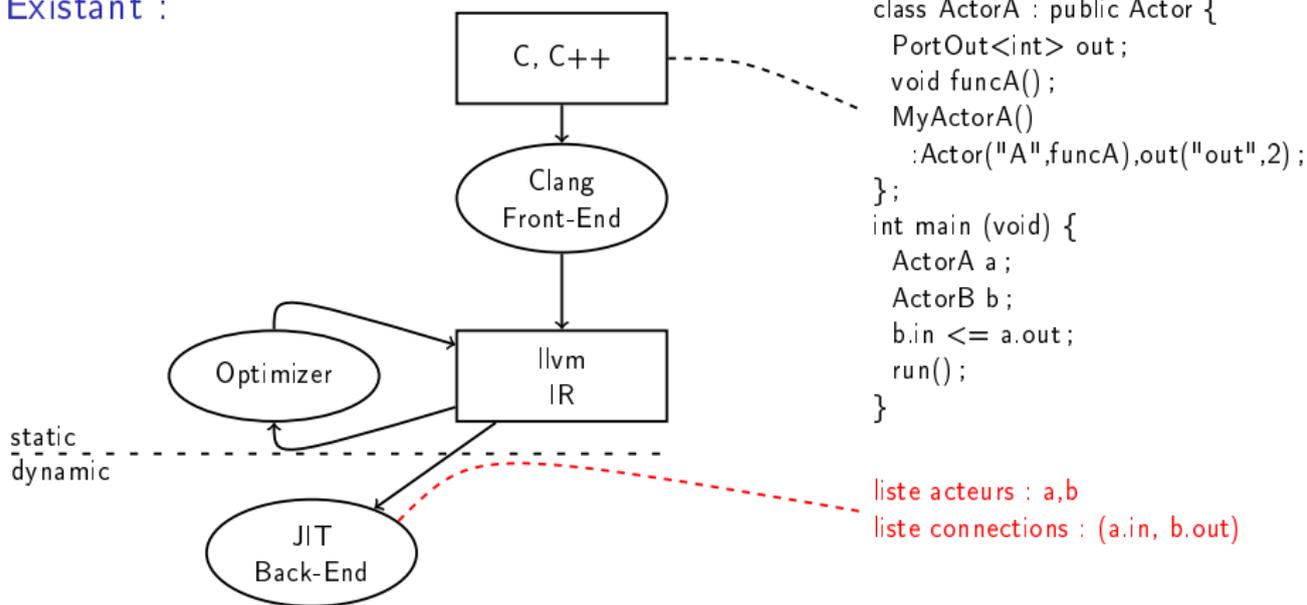
# Front-End

Existant :



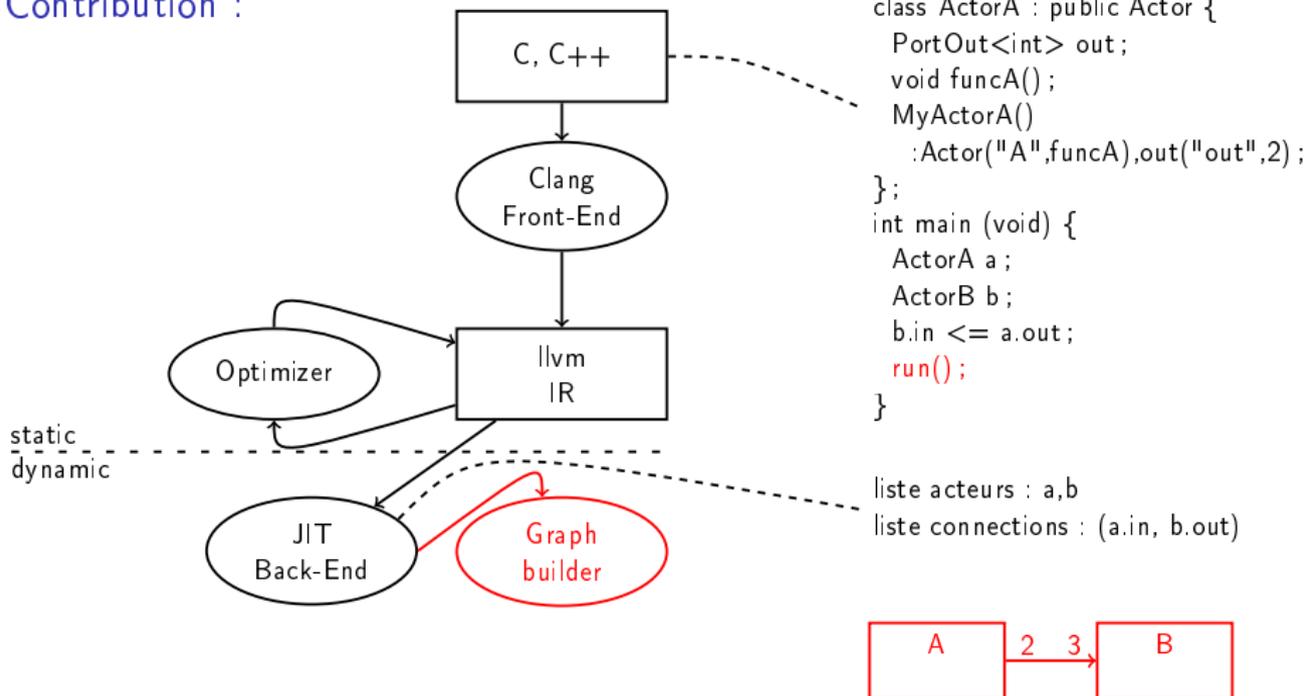
# Front-End

Existant :



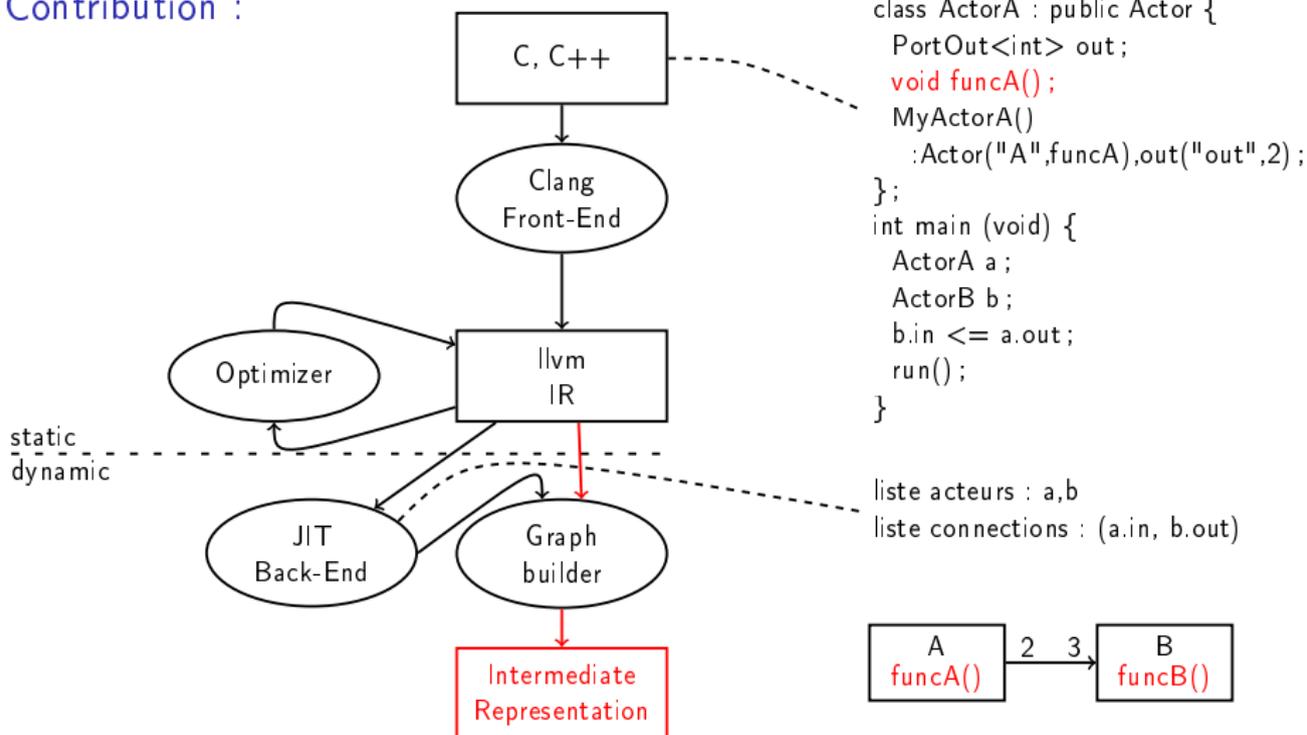
# Front-End

## Contribution :



# Front-End

## Contribution :

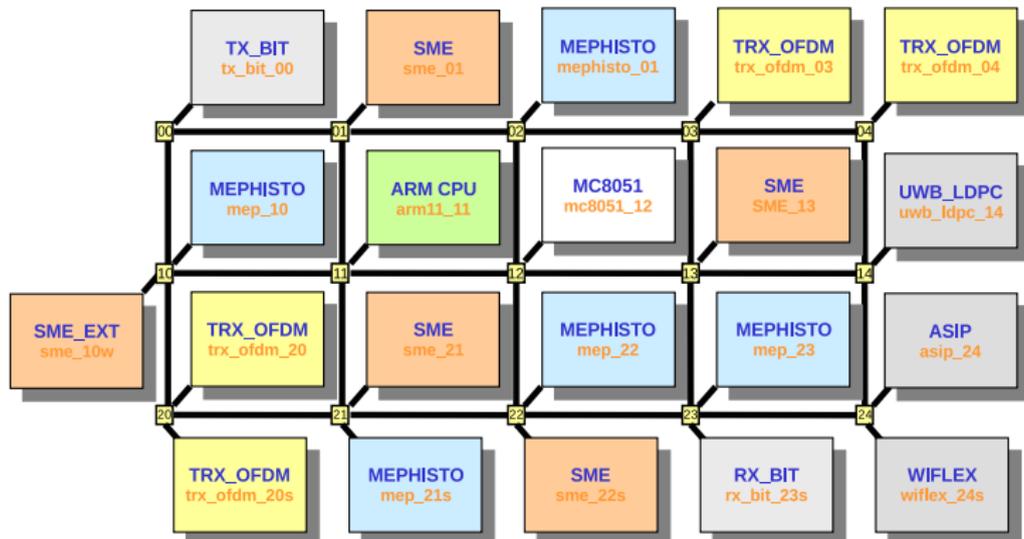


# Plan

- 1 Problématique
  - Application 3GPP/LTE
  - Modèle de calcul
- 2 Chaîne de compilation
  - Infrastructure llvm
  - Front-End
- 3 Plateforme Magali
- 4 Conclusion

## Magali, CEA LETI

- Démonstrateur 3GPP/LTE
- Mémoire distribuée
- Hétérogène
- Réseau sur puce (NoC)
- Technologie 65nm, 2009
- Consommation 231mW



# Problèmes à adresser

## Mapping

- Plateforme hétérogène
- Accélérateurs matériels

## Ordonnancement

- Basé sur SPDF
- Contraintes temps réel

## Génération des communications

- Mécanisme spécifique à Magali
- Support paramètres

# Plan

- 1 Problématique
  - Application 3GPP/LTE
  - Modèle de calcul
- 2 Chaîne de compilation
  - Infrastructure llvm
  - Front-End
- 3 Plateforme Magali
- 4 Conclusion

# Conclusion

## Contribution :

- Langage pour SPDF
- Front-End

## À venir :

- Mapping
- Ordonnancement
- Génération de code
  - Communications
  - Calculs

