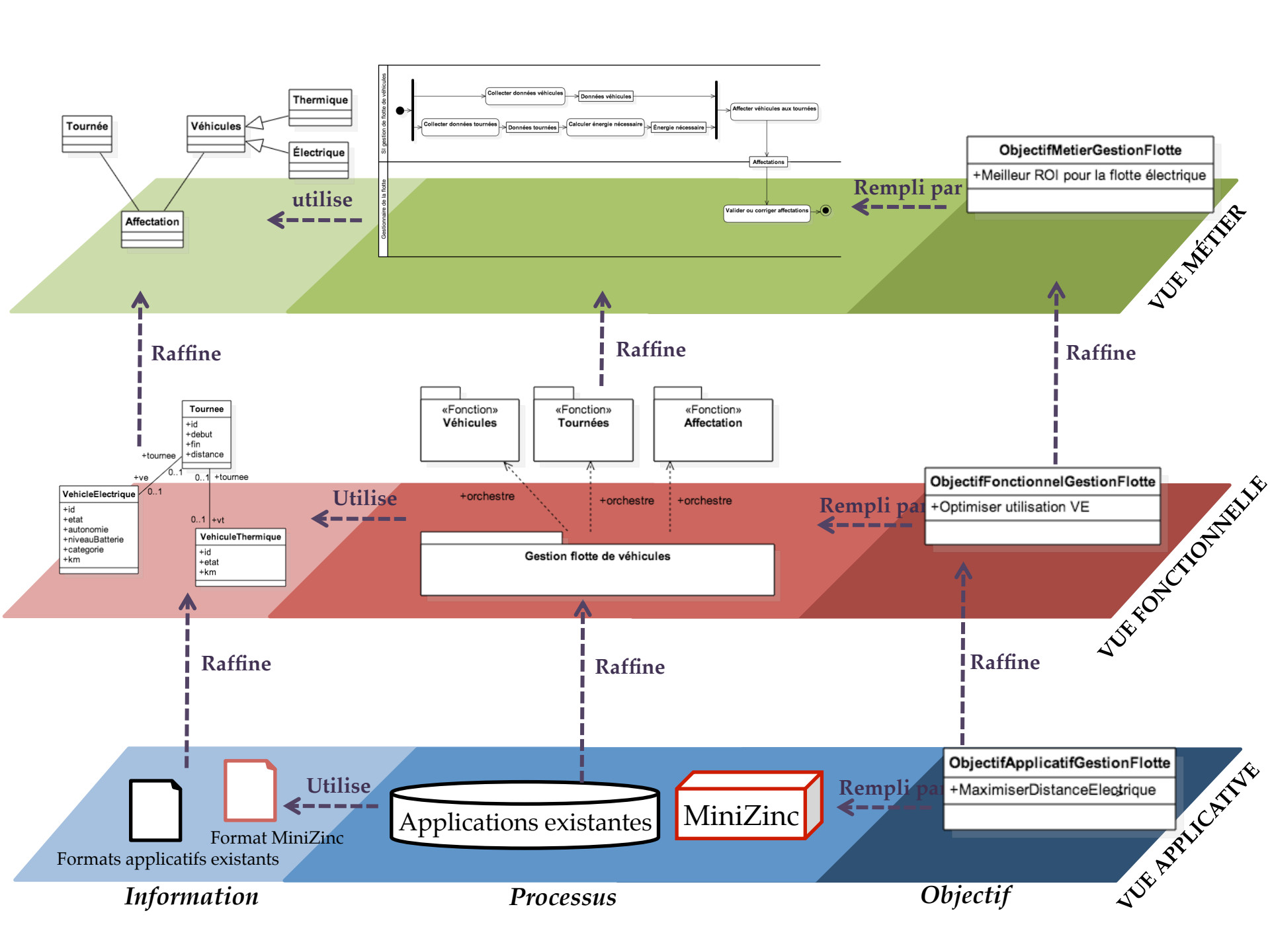
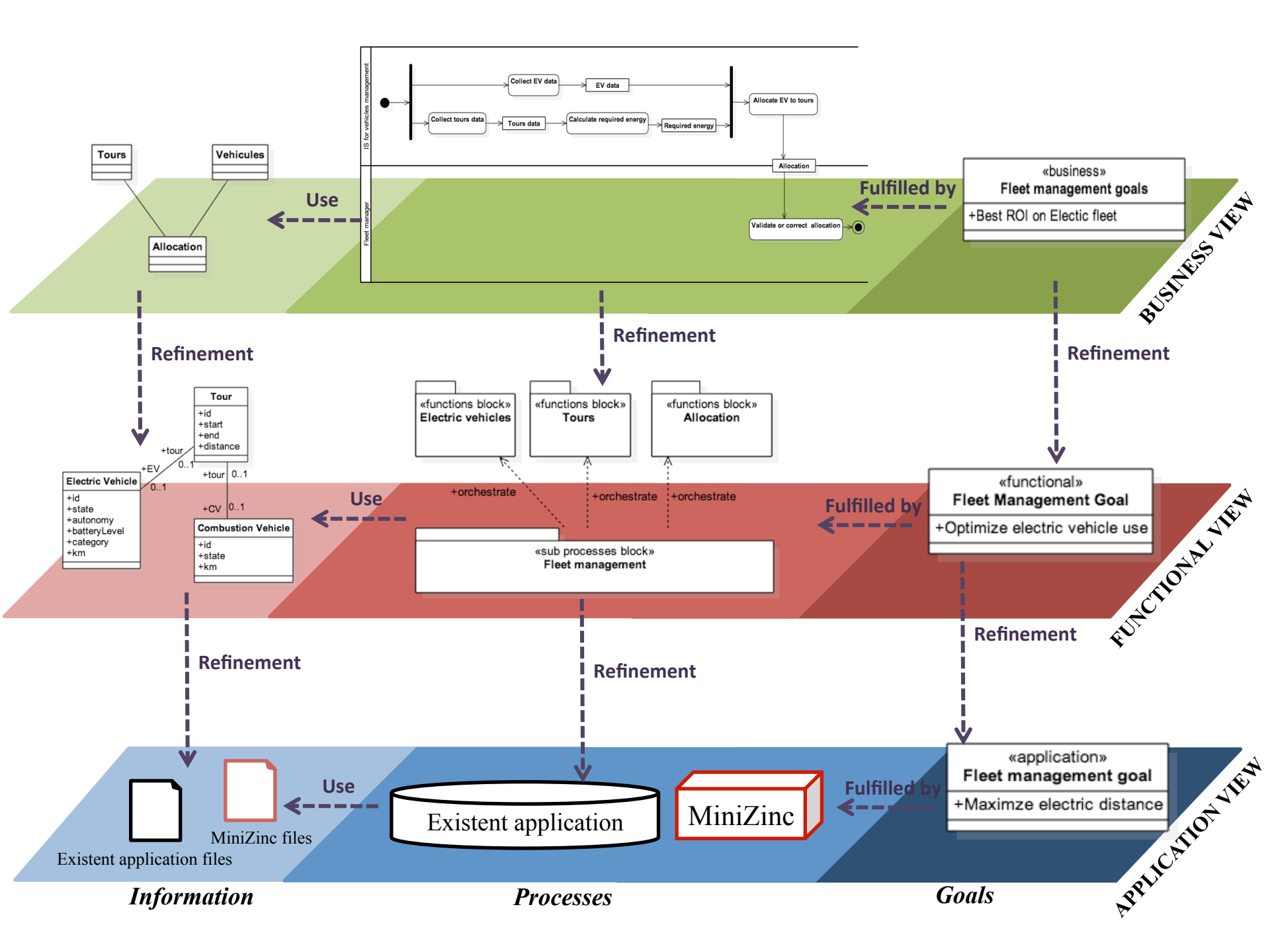
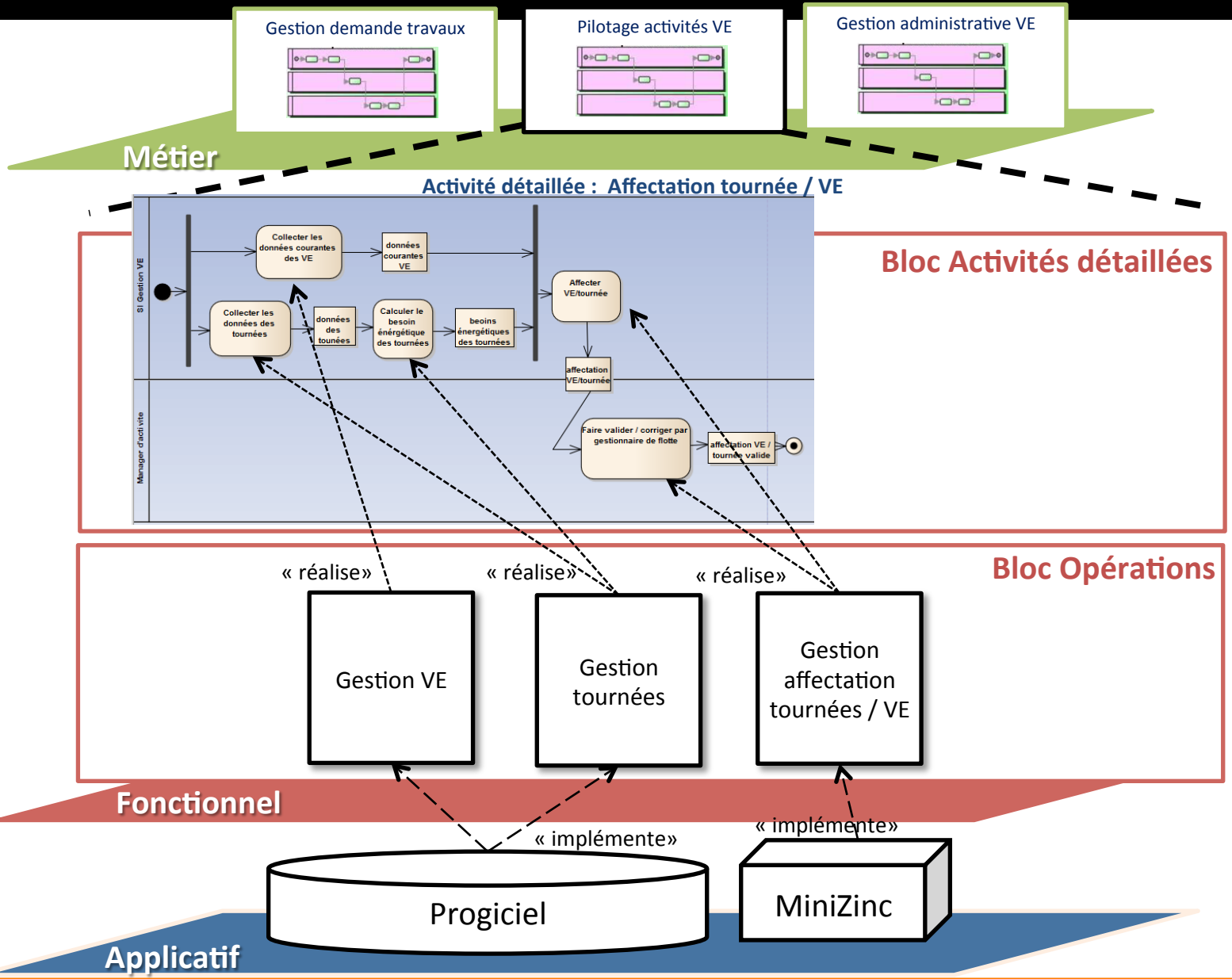
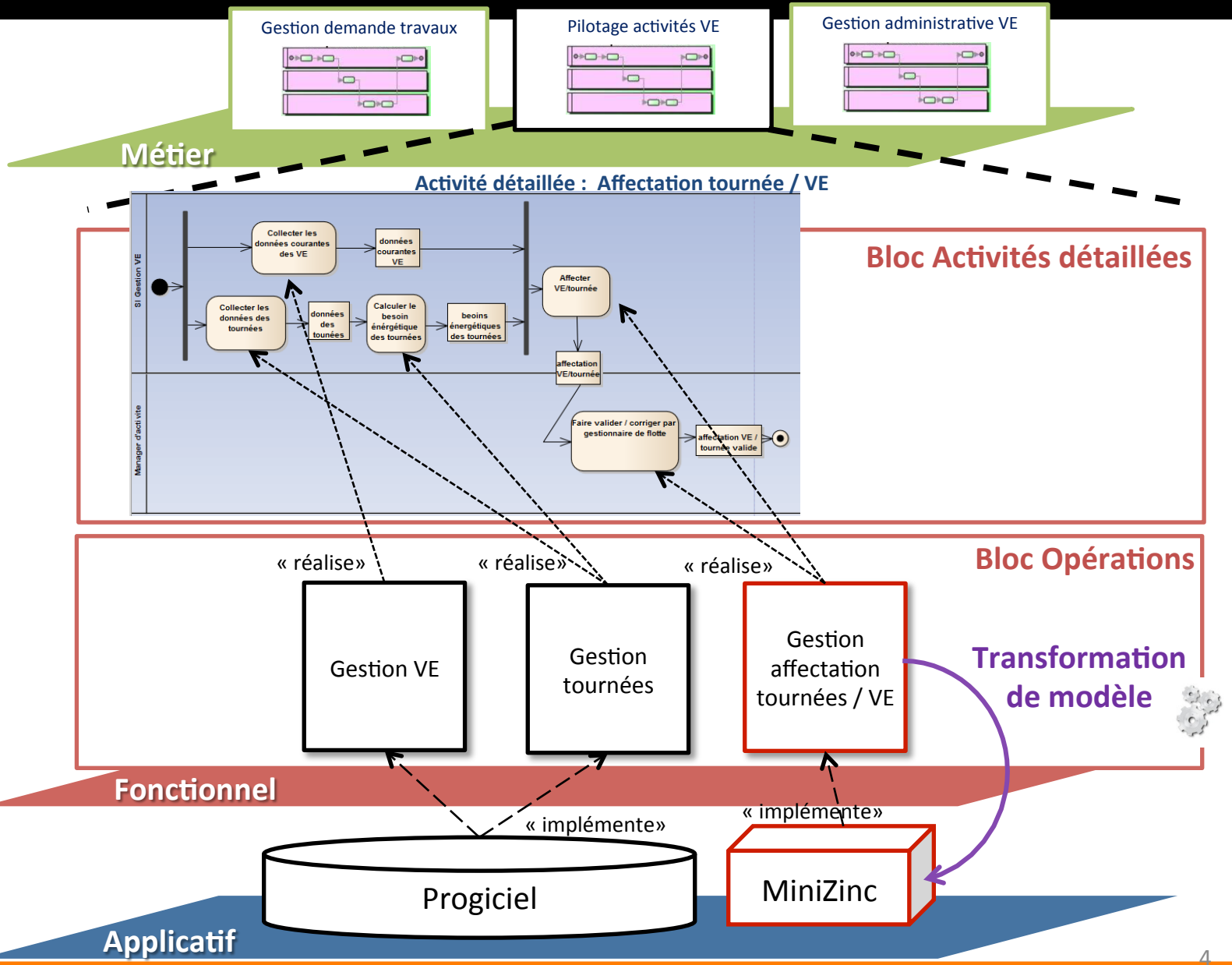

Tournée
Véhicules
Thermique
Électrique
Affectation
utilise
SI gestion de flotte de véhicules
Collecter données véhicules
Données véhicules
Collecter données tournées
Données tournées
Calculer énergie nécessaire
Énergie nécessaire
Affecter véhicules aux tournées
Affectations
Gestionnaire de la flotte
Valider ou corriger affectations
ObjectifMetierGestionFlotte
+Meilleur ROI pour la flotte électrique
Rempli par
VUE MÉTIER
Raffine
Raffine
Raffine
Tournee
+id
+debut
+fin
+distance
+tournee
+ve
0..1
0..1
+tournee
VehicleElectrique
+id
+etat
+autonomie
+niveauBatterie
+categorie
+km
0..1
0..1
+vt
VehiculeThermique
+id
+etat
+km
Utilise
«Fonction»
Véhicules
«Fonction»
Tournées
«Fonction»
Affectation
+orchestre
+orchestre
+orchestre
Gestion flotte de véhicules
ObjectifFonctionnelGestionFlotte
+Optimiser utilisation VE
Rempli par
VUE FONCTIONNELLE
Raffine
Raffine
Raffine
Utilise
Format MiniZinc
Formats applicatifs existants
Applications existantes
MiniZinc
Rempli par
ObjectifApplicatifGestionFlotte
+MaximiserDistanceElectrique
VUE APPLICATIVE
Information
Processus
Objectif


IS for vehicles management
Fleet manager
Collect EV data
EV data
Collect tours data
Tours data
Calculate required energy
Required energy
Allocate EV to tours
Allocation
Validate or correct allocation
Tours
Vehicules
Allocation
Use
Fulfilled by
«business»
Fleet management goals
+Best ROI on Electic fleet
BUSINESS VIEW
Refinement
Refinement
Refinement
Tour
+id
+start
+end
+distance
+tour
+EV
+tour
+CV
0..1
Electric Vehicle
+id
+state
+autonomy
+batteryLevel
+category
+km
Combustion Vehicle
+id
+state
+km
«functions block»
Electric vehicles
«functions block»
Tours
«functions block»
Allocation
+orchestrate
+orchestrate
+orchestrate
Use
«sub processes block»
Fleet management
Fulfilled by
«functional»
Fleet Management Goal
+Optimize electric vehicle use
FUNCTIONAL VIEW
Refinement
Refinement
Refinement
Use
Existent application files
MiniZinc files
Existent application
MiniZinc
Fulfilled by
«application»
Fleet management goal
+Maximze electric distance
APPLICATION VIEW
Information
Processes
Goals

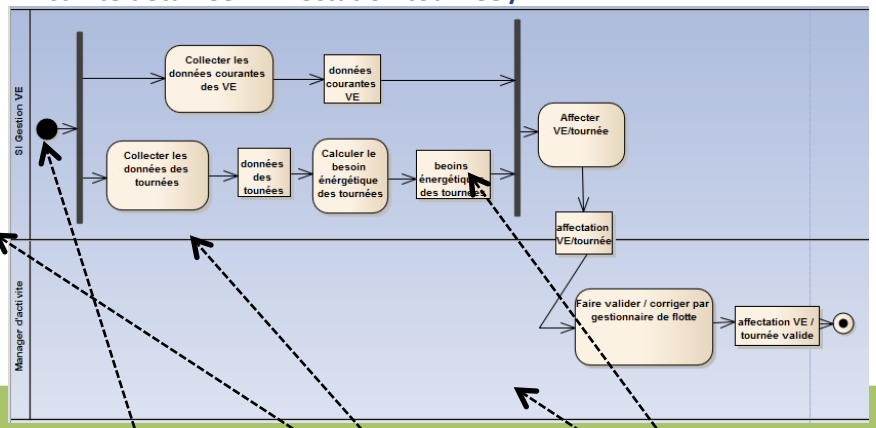
Mise en œuvre pour le cas métier



Mise en œuvre pour le cas métier



Activité détaillée : Affectation tournée / VE



Métier

« réalise»

« réalise»

« réalise»

Bloc Opérations

Transformation de modèle

Gestion VE

Gestion tournées

Gestion affectation tournées / VE

Fonctionnel

« implémente»

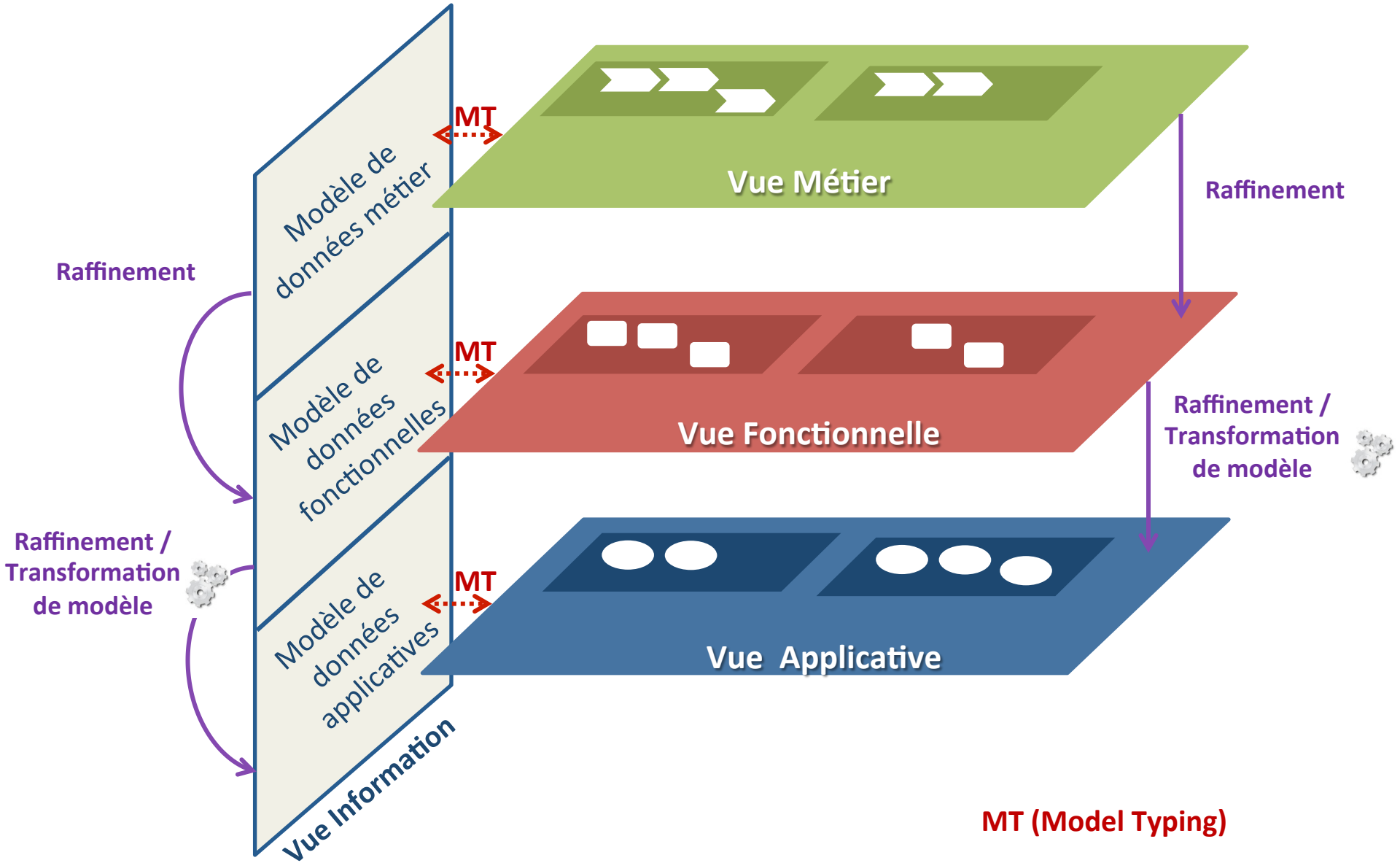
« implémente»

Progiciel

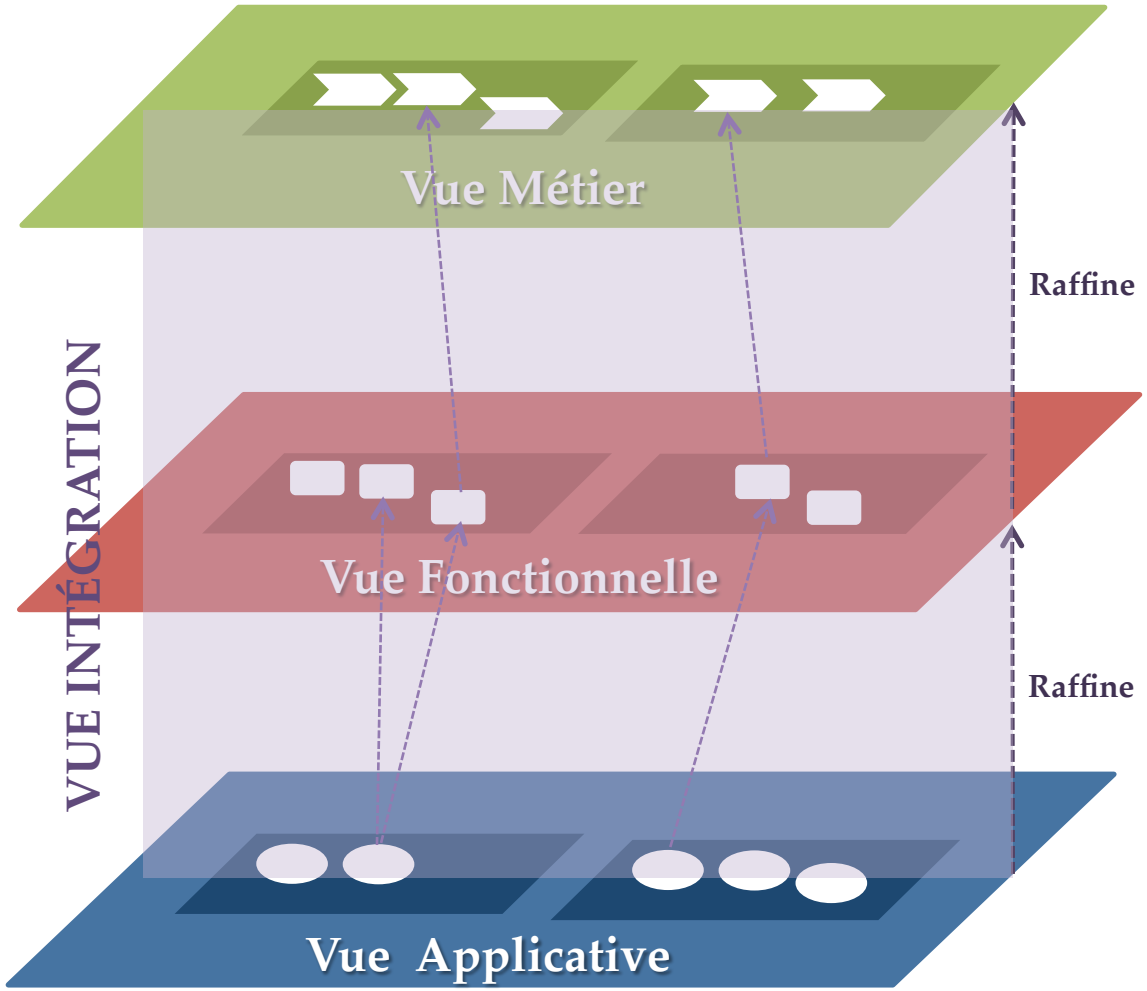
MiniZinc

Applicatif

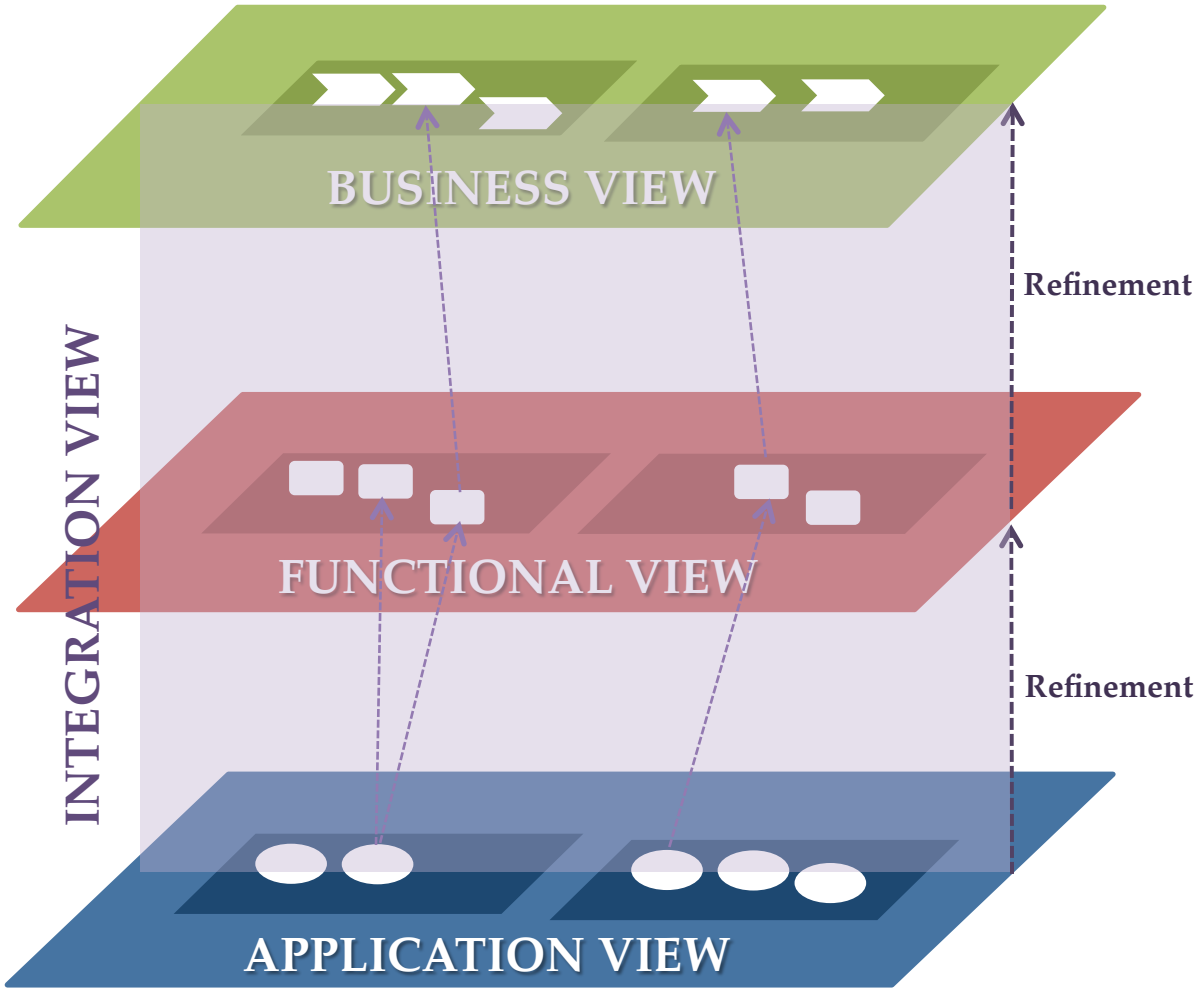
Démarche


Modèle de données métier
Modèle de données fonctionnelles
Modèle de données applicatives
Vue Information
MT
MT
MT
Vue Métier
Vue Fonctionnelle
Vue Applicative
Raffinement
Raffinement
Raffinement / Transformation de modèle
Raffinement / Transformation de modèle
MT (Model Typing)

Démarche


Vue Métier
Raffine
VUE INTÉGRATION
Vue Fonctionnelle
Raffine
Vue Applicative

Démarche

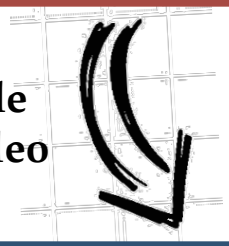

BUSINESS VIEW
FUNCTIONAL VIEW
APPLICATION VIEW
INTEGRATION VIEW
Refinement
Refinement

Vue fonctionnelle

Gestion affectation

```
context Tournee
inv:    self.VE.autonomie * self.VE.niveauBatterie > self.distanceParcourue

context Tournee
inv:    self.VE <> undefined xor self.VT <> undefined

context Tournee::kmElec(): int
body:   (Tournee::allInstances()->collect(t.VE <> undefined |t.distanceParcourue))->sum()
```

Transformation de modèle avec Acceleo



Vue applicative

MiniZinc

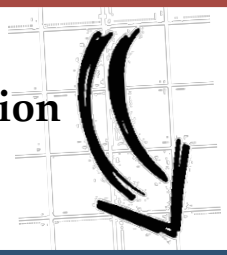
```
constraint forall (i in Tournees) (
%affectation de vehicule electrique si l autonomie l autorise
constraint forall(i in  Tournees, j in VehiculesElec)
(tourneeVehicule[i]= identifiantVE[j] -> (autonomie[j]*niveauBatterie[j] > distanceTournee[i] /\
kmElec[i]=distanceTournee[i]));

% affectation d un vehicule thermique et dans ce cas kmElec est nul
constraint forall(i in  Tournees, k in VehiculesTherm)
(tourneeVehicule[i]= identifiantVT[k] -> kmElec[i]=0);

%maximiser le nombre de km de tounrnee fait par les vehicules electriques
solve maximize sum(i in Tournees)(kmElec[i]);
```

Functional view

Allocation function

```
context Tour
inv : self.EV.autonomy * self.EV.batteryLevel > self.neededEnergy

context Tour
inv : self.EV <> undefined xor self.CV <> undefined

context Tour :: elecKm() : int
body :  (Tours::allInstances() -> collect(t.EV <> undefined |t.distance)) ->
sum()
```

Model transformation with Acceleo

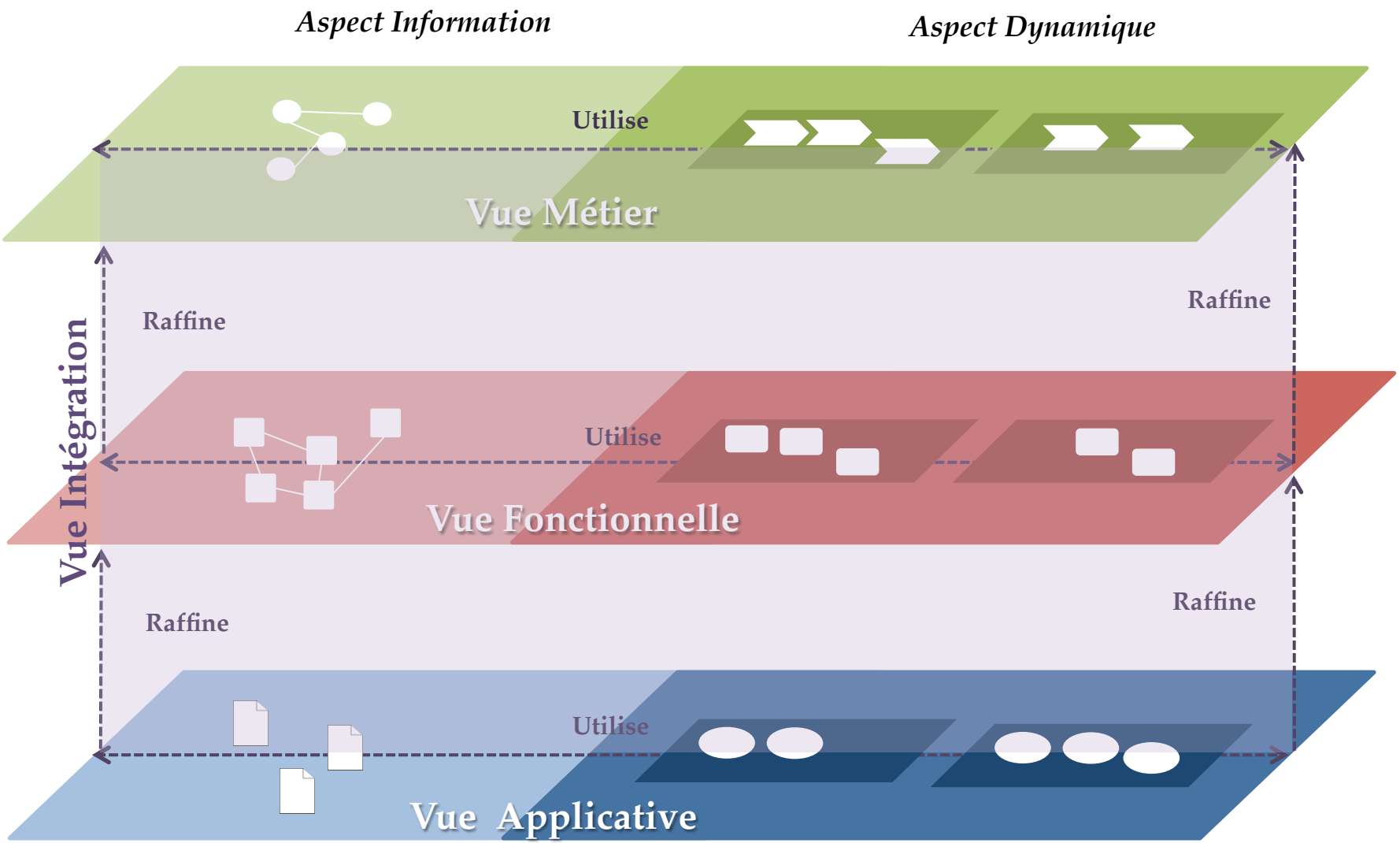


Application view

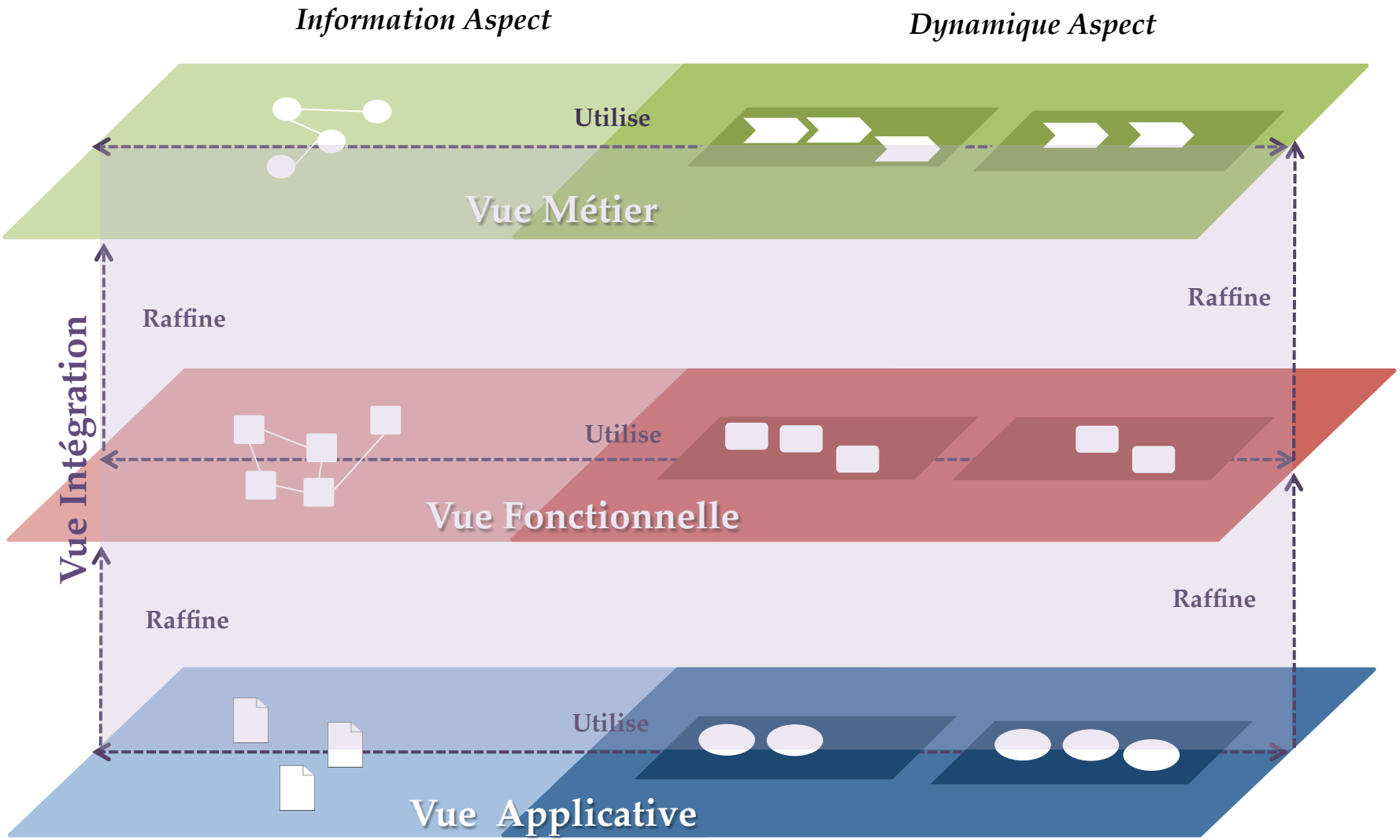
MiniZinc Module

```
%a vehicle is affected to only one tour
constraint alldifferent(allocation);

%allocate an electric vehicle if the autonomy permits it
constraint forall(i in  Tours, j in ElectricVehicles)
(allocation[i]= id_EV[j] -> (autonomy[j] > neededEnergy_Tours[i]
/\ kmElec[i]= distance_Tours[i]));

% allocate a combustion vehicle
constraint forall(i in  Tours, k in CombustionVehicles)
(allocation[i]= id_CV[k] -> kmElec[i]=0);

%maximze km done by electric vehicles
solve maximize sum(i in Tours)(kmElec[i]);'
```

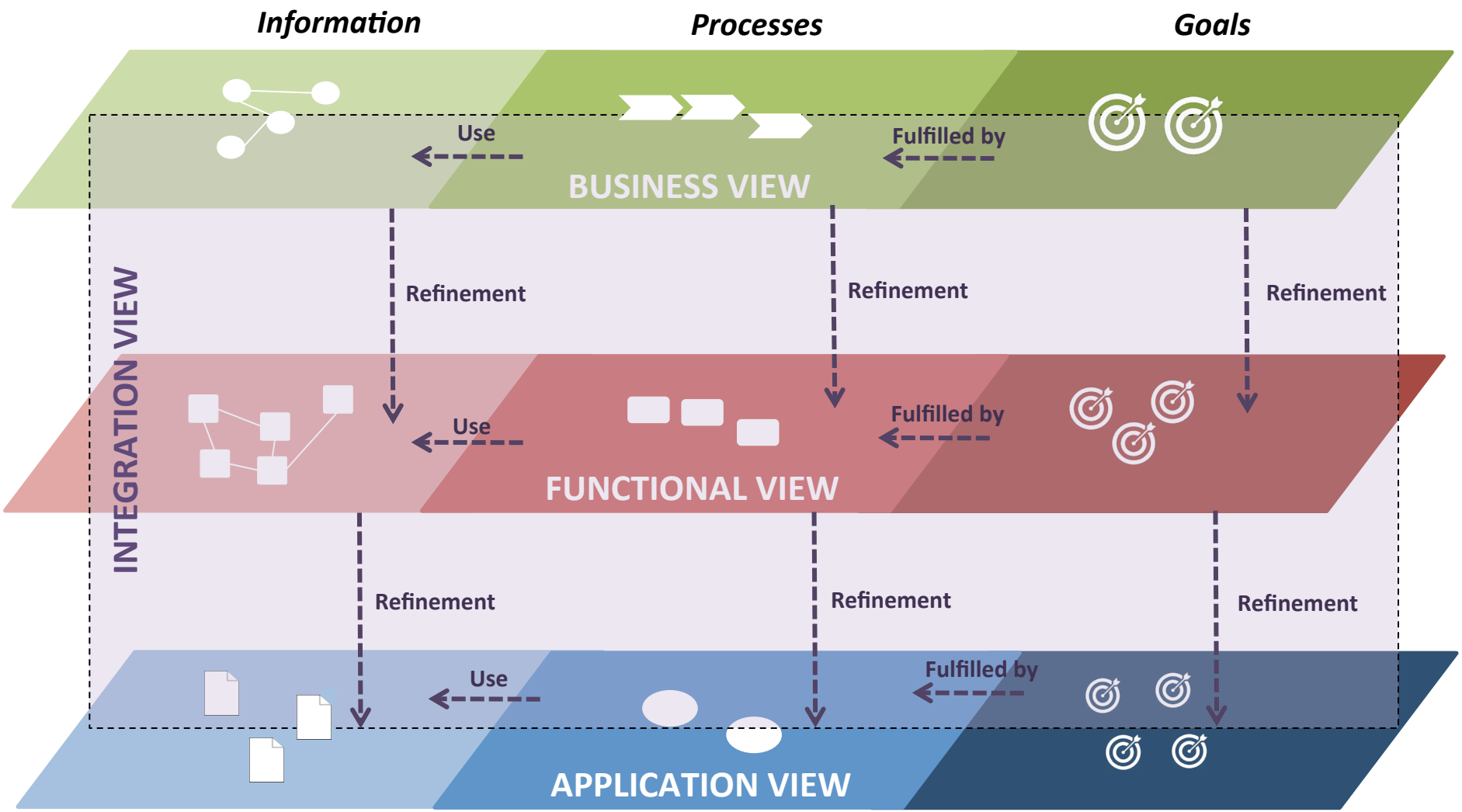
Démarche


Aspect Information
Aspect Dynamique
Utilise
Vue Métier
Raffine
Raffine
Vue Intégration
Utilise
Vue Fonctionnelle
Raffine
Raffine
Utilise
Vue Applicative

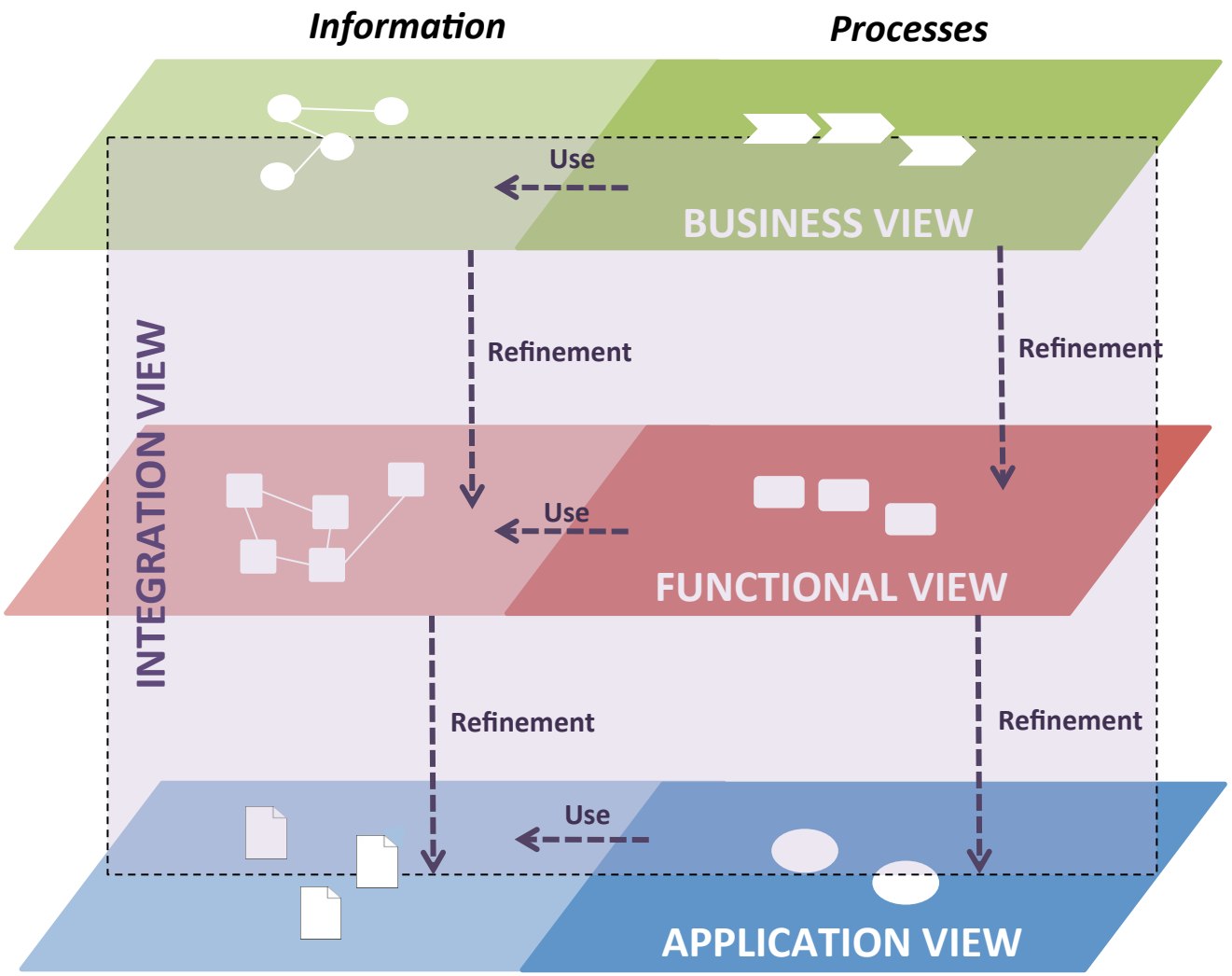
Approach

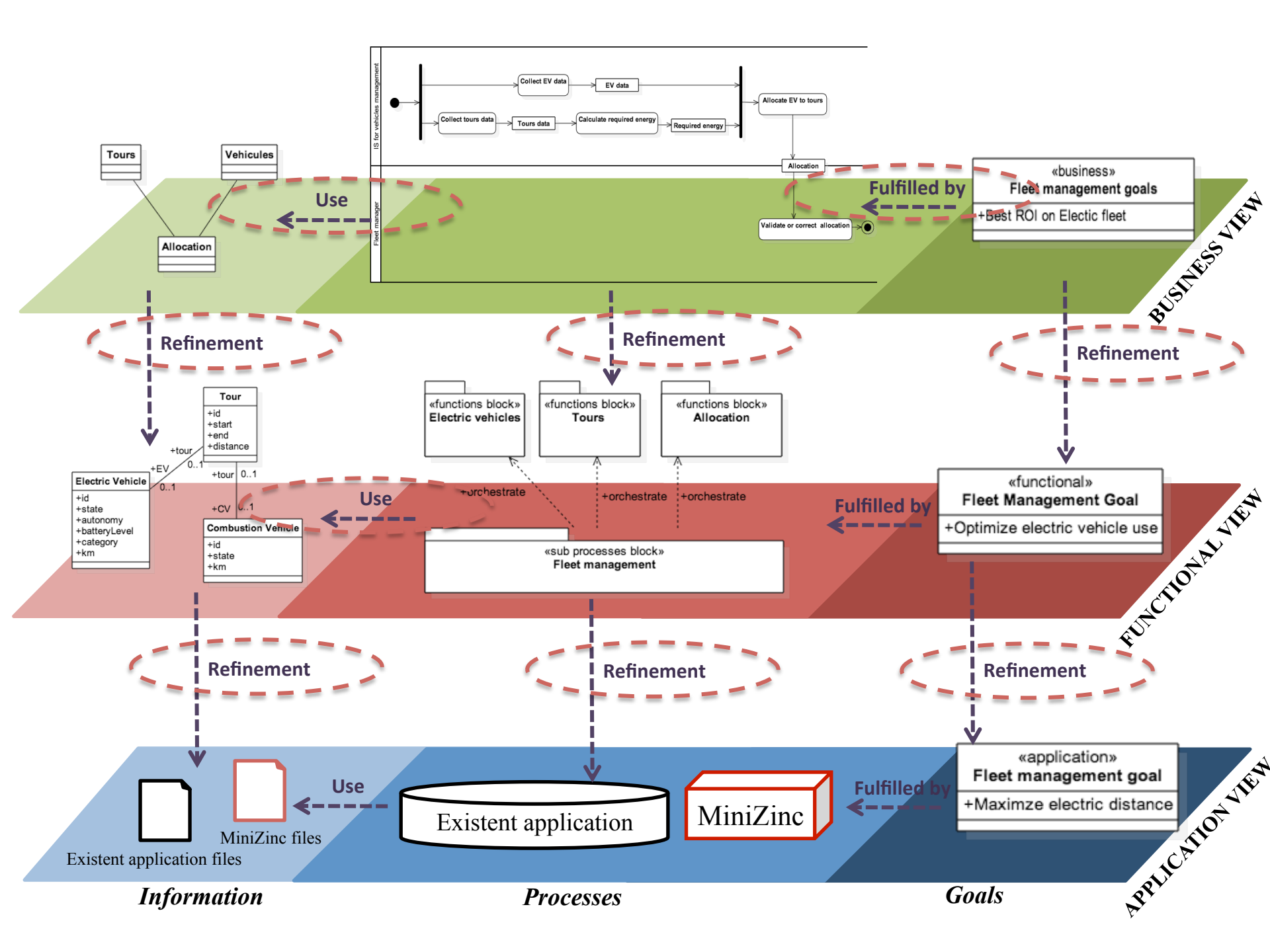

Information Aspect
Dynamique Aspect
Utilise
Vue Métier
Raffine
Raffine
Vue Intégration
Utilise
Vue Fonctionnelle
Raffine
Raffine
Utilise
Vue Applicative

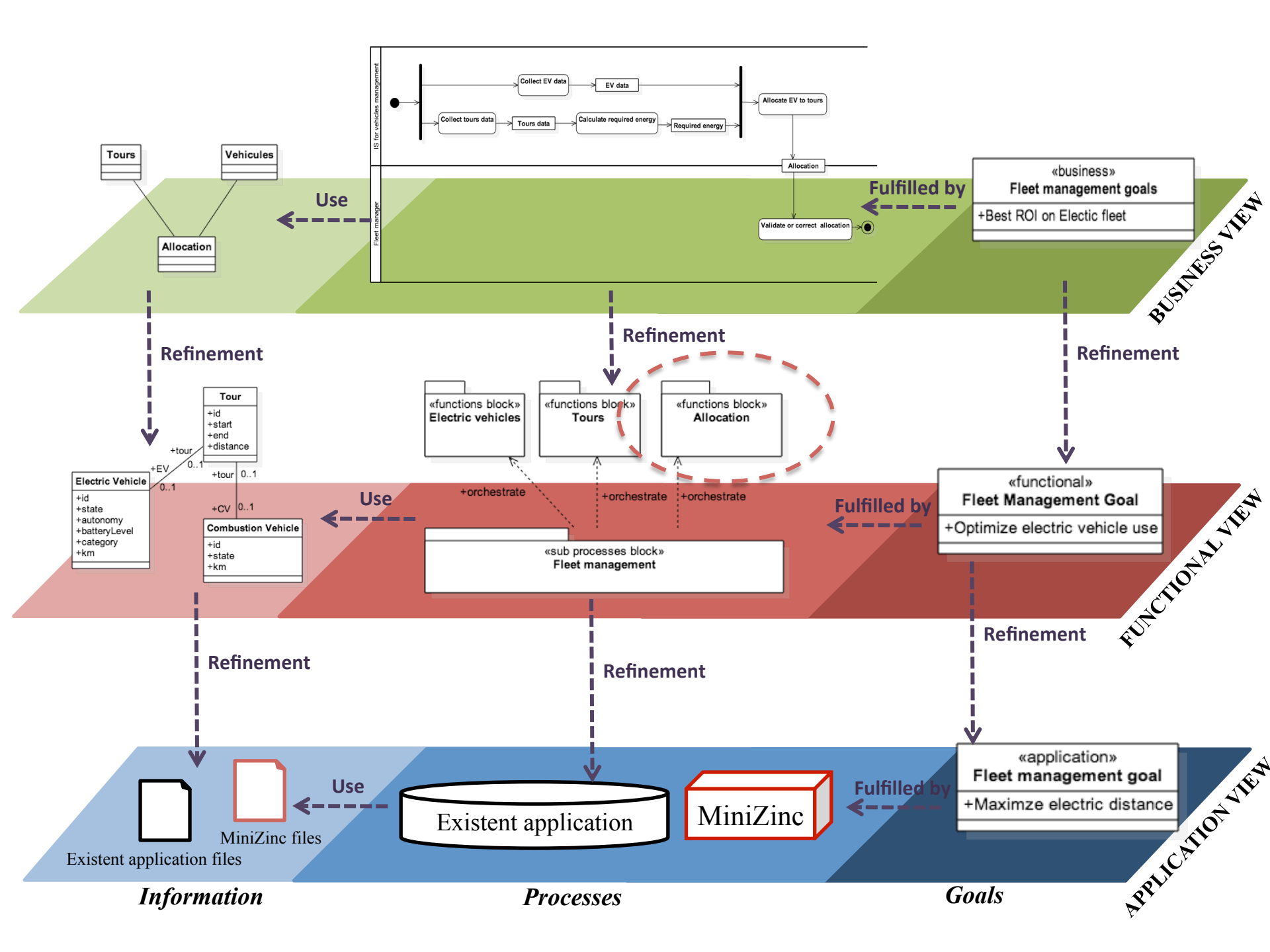
Démarche

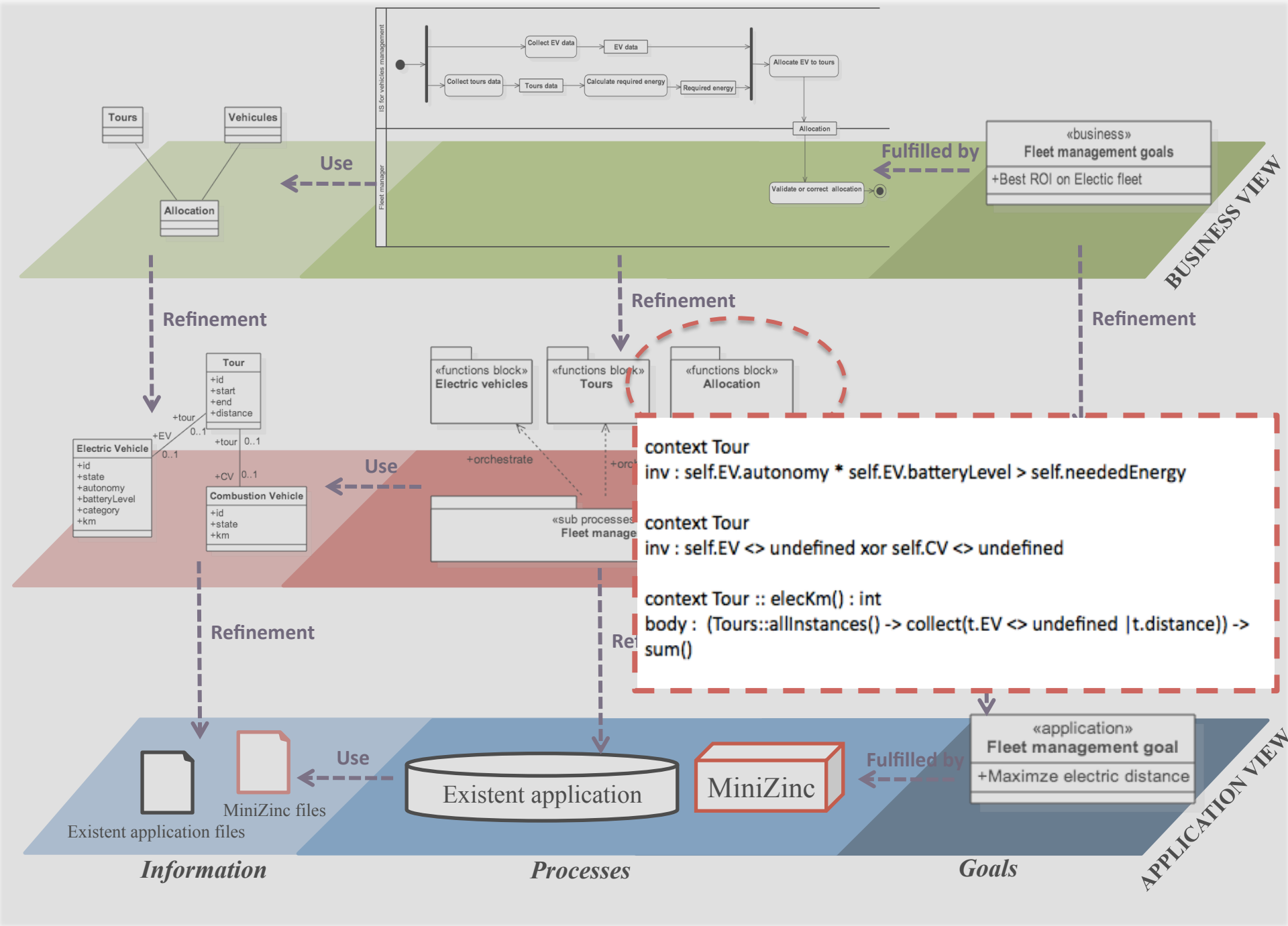

Information
Processes
Goals
Use
Fulfilled by
BUSINESS VIEW
Refinement
Refinement
Refinement
INTEGRATION VIEW
Use
Fulfilled by
FUNCTIONAL VIEW
Refinement
Refinement
Refinement
Use
Fulfilled by
APPLICATION VIEW

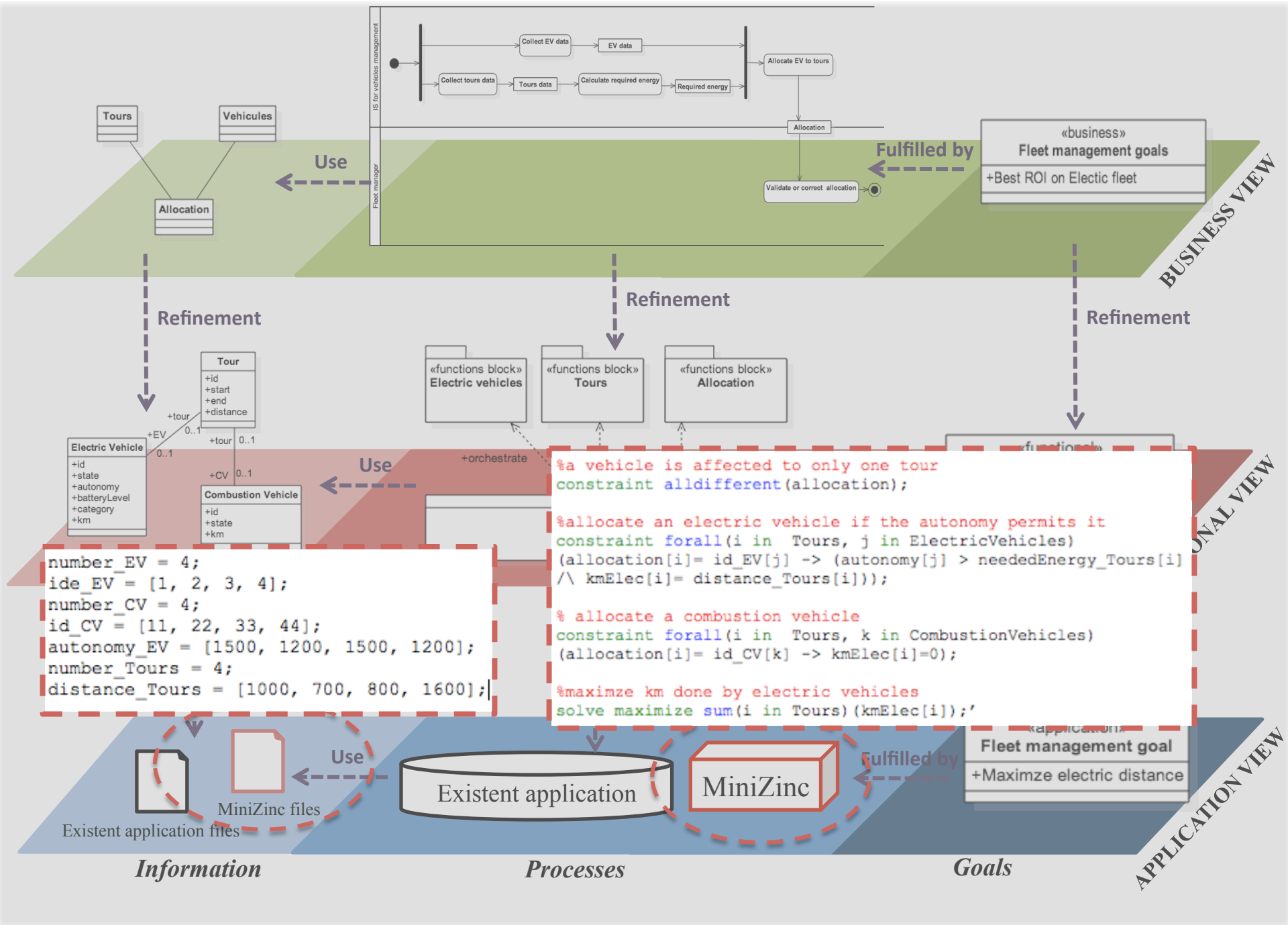
Démarche

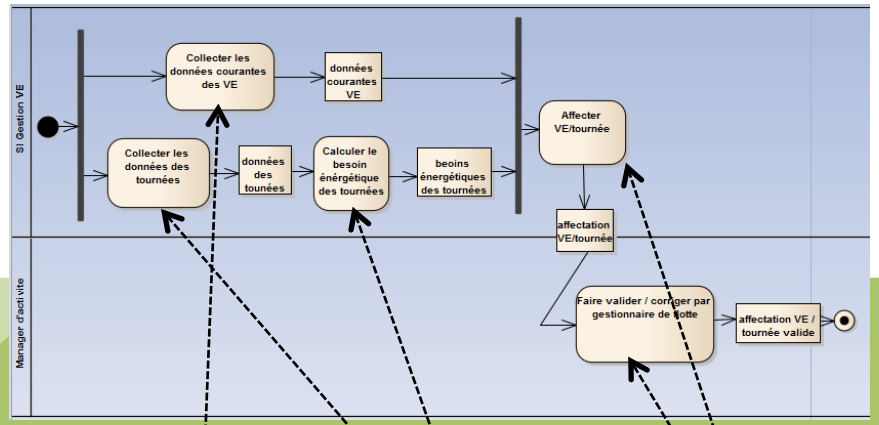

Information
Processes
Use
BUSINESS VIEW
Refinement
Refinement
INTEGRATION VIEW
Use
FUNCTIONAL VIEW
Refinement
Refinement
Use
APPLICATION VIEW


IS for vehicles management
Fleet manager
Collect EV data
EV data
Collect tours data
Tours data
Calculate required energy
Required energy
Allocate EV to tours
Allocation
Validate or correct allocation
Tours
Vehicules
Allocation
Use
Fulfilled by
«business»
Fleet management goals
+Best ROI on Electic fleet
BUSINESS VIEW
Refinement
Refinement
Refinement
Tour
+id
+start
+end
+distance
+tour
0..1
+EV
0..1
+tour
0..1
Electric Vehicle
+id
+state
+autonomy
+batteryLevel
+category
+km
+CV
0..1
Combustion Vehicle
+id
+state
+km
«functions block»
Electric vehicles
«functions block»
Tours
«functions block»
Allocation
Use
+orchestrate
+orchestrate
+orchestrate
«sub processes block»
Fleet management
Fulfilled by
«functional»
Fleet Management Goal
+Optimize electric vehicle use
FUNCTIONAL VIEW
Refinement
Refinement
Refinement
Use
Existent application files
MiniZinc files
Existent application
MiniZinc
Fulfilled by
«application»
Fleet management goal
+Maximze electric distance
APPLICATION VIEW
Information
Processes
Goals


IS for vehicles management
Fleet manager
Collect EV data
EV data
Collect tours data
Tours data
Calculate required energy
Required energy
Allocate EV to tours
Allocation
Validate or correct allocation
Tours
Vehicules
Allocation
Use
Fulfilled by
«business»
Fleet management goals
+Best ROI on Electic fleet
BUSINESS VIEW
Refinement
Refinement
Refinement
Tour
+id
+start
+end
+distance
+tour
0..1
+EV
0..1
+tour
0..1
Electric Vehicle
+id
+state
+autonomy
+batteryLevel
+category
+km
+CV
0..1
Combustion Vehicle
+id
+state
+km
«functions block»
Electric vehicles
«functions block»
Tours
«functions block»
Allocation
Use
+orchestrate
+orchestrate
+orchestrate
«sub processes block»
Fleet management
Fulfilled by
«functional»
Fleet Management Goal
+Optimize electric vehicle use
FUNCTIONAL VIEW
Refinement
Refinement
Refinement
Use
Existent application files
MiniZinc files
Existent application
MiniZinc
Fulfilled by
«application»
Fleet management goal
+Maximze electric distance
APPLICATION VIEW
Information
Processes
Goals


IS for vehicles management
Fleet manager
Collect EV data
EV data
Collect tours data
Tours data
Calculate required energy
Required energy
Allocate EV to tours
Allocation
Validate or correct allocation
Tours
Vehicules
Allocation
Use
Fulfilled by
«business»
Fleet management goals
+Best ROI on Electic fleet
BUSINESS VIEW
Refinement
Refinement
Refinement
Tour
+id
+start
+end
+distance
+tour
0..1
+EV
0..1
+tour
0..1
Electric Vehicle
+id
+state
+autonomy
+batteryLevel
+category
+km
+CV
0..1
Combustion Vehicle
+id
+state
+km
«functions block»
Electric vehicles
«functions block»
Tours
«functions block»
Allocation
Use
+orchestrate
«sub processes
Fleet manage
context Tour
inv : self.EV.autonomy * self.EV.batteryLevel > self.neededEnergy
context Tour
inv : self.EV <> undefined xor self.CV <> undefined
context Tour :: elecKm() : int
body : (Tours::allInstances() -> collect(t.EV <> undefined |t.distance)) ->
sum()
Refinement
Use
Existent application
MiniZinc
Fulfilled by
«application»
Fleet management goal
+Maximze electric distance
APPLICATION VIEW
MiniZinc files
Existent application files
Information
Processes
Goals


Tours
Vehicules
Allocation
Use
IS for vehicles management
Fleet manager
Collect EV data
EV data
Collect tours data
Tours data
Calculate required energy
Required energy
Allocate EV to tours
Allocation
Validate or correct allocation
Fulfilled by
«business»
Fleet management goals
+Best ROI on Electic fleet
BUSINESS VIEW
Refinement
Refinement
Refinement
Tour
+id
+start
+end
+distance
+tour
0..1
+EV
+tour
0..1
Electric Vehicle
+id
+state
+autonomy
+batteryLevel
+category
+km
+CV
0..1
Combustion Vehicle
+id
+state
+km
«functions block»
Electric vehicles
«functions block»
Tours
«functions block»
Allocation
Use
+orchestrate
%a vehicle is affected to only one tour
constraint alldifferent(allocation);
%allocate an electric vehicle if the autonomy permits it
constraint forall(i in Tours, j in ElectricVehicles)
(allocation[i]= id_EV[j] -> (autonomy[j] > neededEnergy_Tours[i]
/\ kmElec[i]= distance_Tours[i]));
% allocate a combustion vehicle
constraint forall(i in Tours, k in CombustionVehicles)
(allocation[i]= id_CV[k] -> kmElec[i]=0);
%maximze km done by electric vehicles
solve maximize sum(i in Tours)(kmElec[i]);'
number_EV = 4;
ide_EV = [1, 2, 3, 4];
number_CV = 4;
id_CV = [11, 22, 33, 44];
autonomy_EV = [1500, 1200, 1500, 1200];
number_Tours = 4;
distance_Tours = [1000, 700, 800, 1600];
Use
MiniZinc files
Existent application files
Existent application
MiniZinc
Fulfilled by
Fleet management goal
+Maximze electric distance
APPLICATION VIEW
Information
Processes
Goals



Vue Métier

« réalise»

« réalise»

« réalise»

Gestion VE

Gestion tournées

Gestion affectation tournées / VE

Vue Fonctionnelle

« implémente»

« implémente»

Progiciel

MiniZinc

Vue Applicative