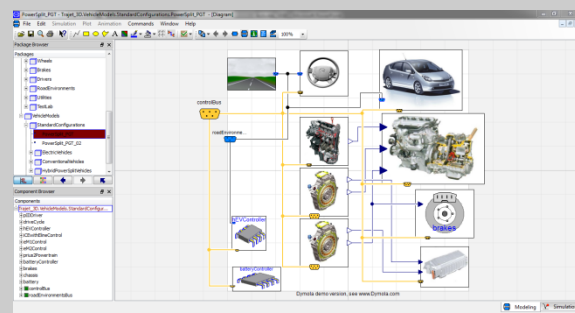
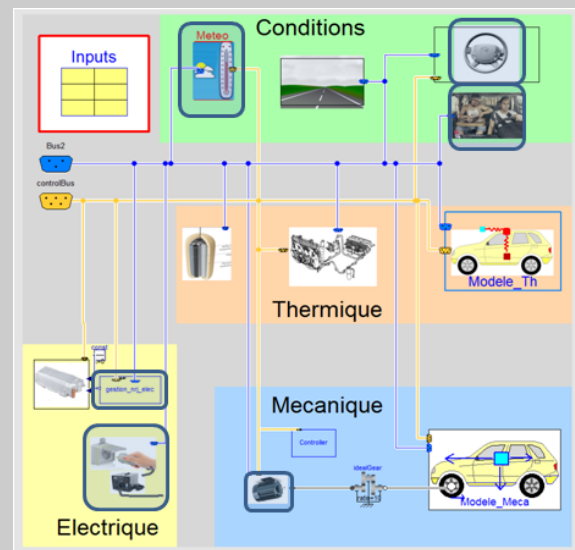
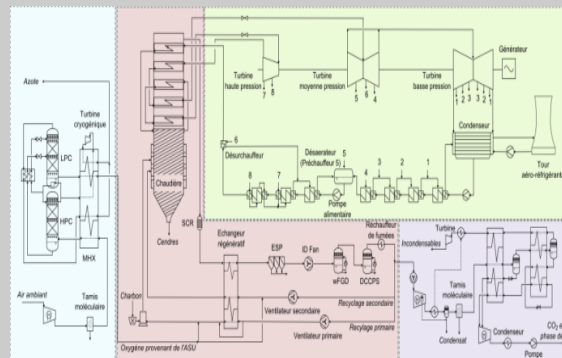


Le CES en quelques mots

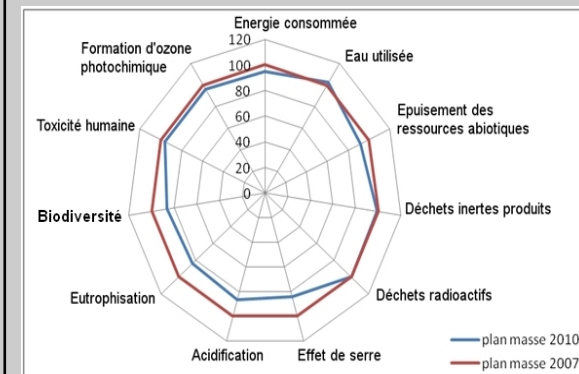
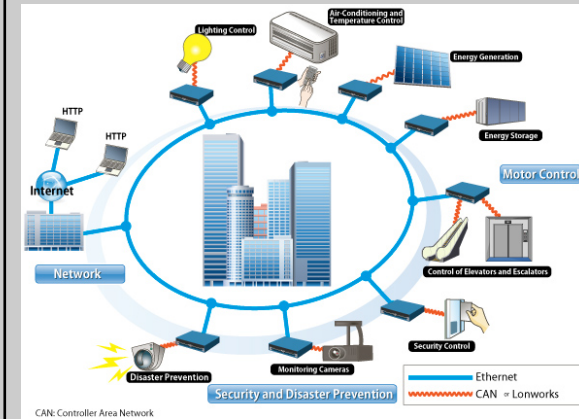
Véhicules basses émissions



Industrie efficace et décarbonée



Bâtiments et villes durables

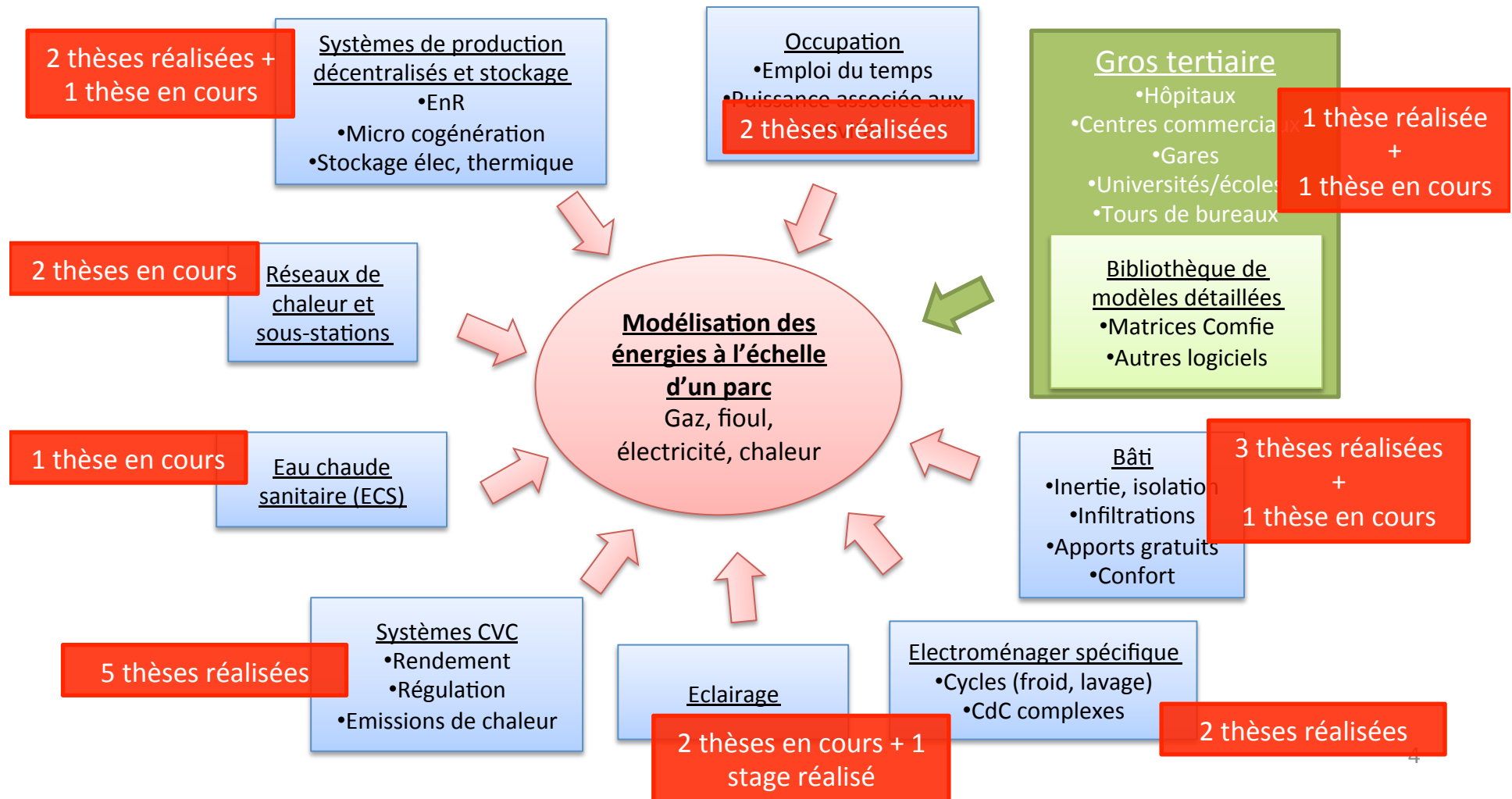
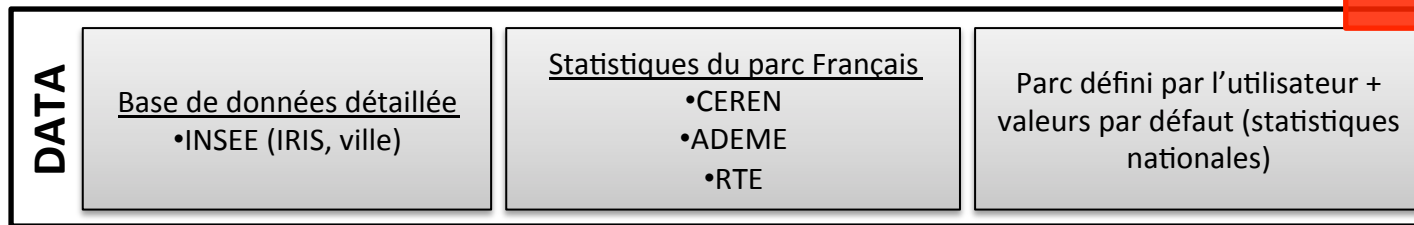


Objectifs du projet Smart-E

- Simulation techno-explicite (bottom-up) des consommations énergétiques de n'importe quelle ville en France ou d'un parc hypothétique – bâtiments et équipements
- Calcul ultra-rapide (~10 000 logements en moins d'une heure)
- Test à l'échelle 1:1 de stratégies de Maîtrise de la Demande Énergétique : rénovation, effacement, évolution des comportements, réseaux intelligents
- Utiliser les modèles du CES : adapter et connecter les modèles existants dans une plateforme informatique

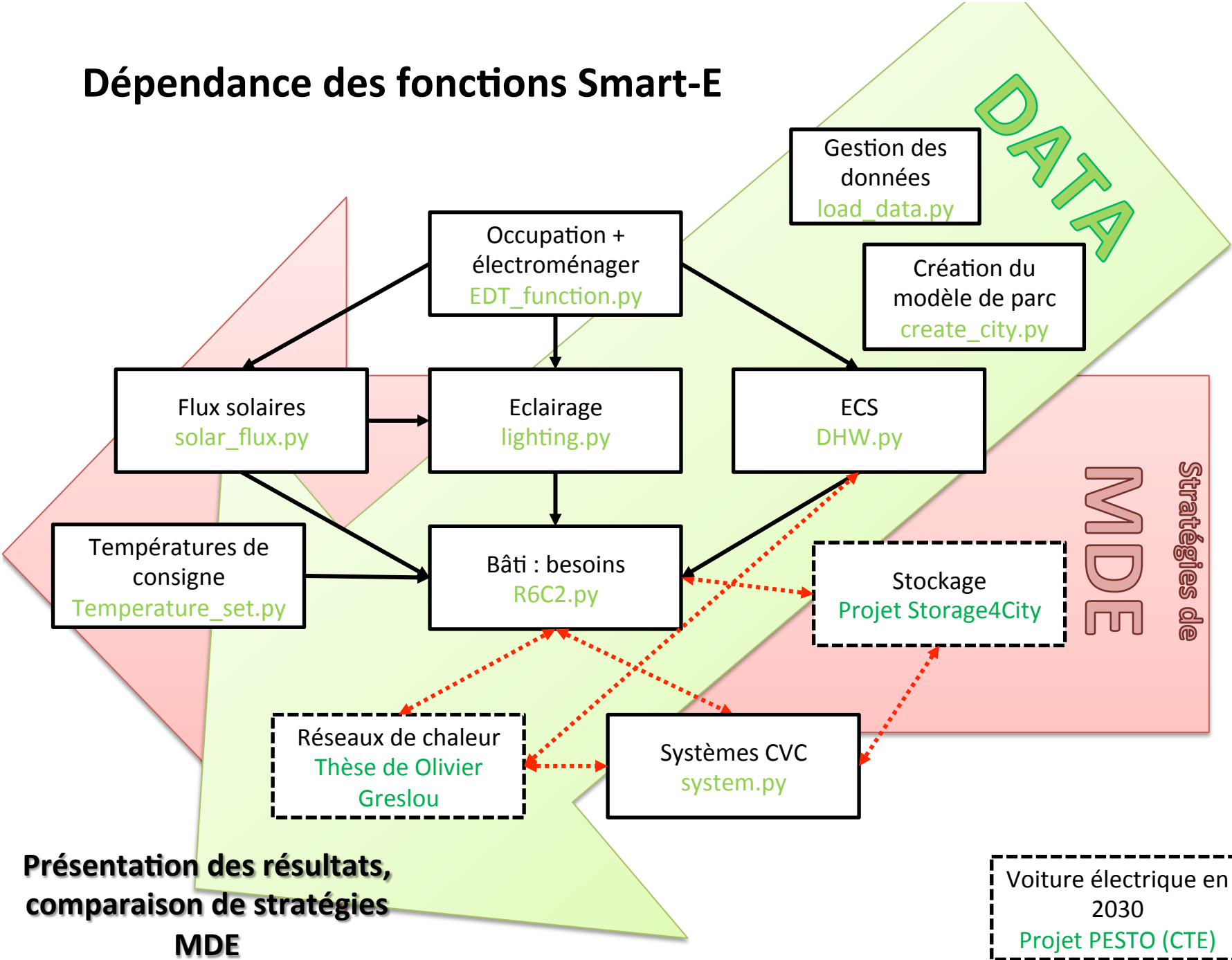
Une plateforme de simulation modulaire qui s'appuie sur les travaux du CES

Etudes MDE depuis 2004



2. ARCHITECTURE DE SMART-E ET PRÉSENTATION DES BASES DE DONNÉES

Dépendance des fonctions Smart-E



Exemple de fonction : lighting.py

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jul 15 14:43:34 2014
Function for lighting consumption calculation (from
RT2005-2012)
RT2005 : ThC-E page 22
I_dir : direct radiation on the windows (W/m2)
I_ref : reflection radiation part on the windows (W/m2)
I_dif_surf : beam radiation on the windows (W/m2)
@author: tberthou
"""
import numpy as np

def lighting(city, occ, rest_resu, I_dir_s, I_dir_n,
            I_dir_e, I_dir_o, I_ref, I_dif_surf, mode):

    coef_shading = 0.5 # à corriger
    b_sur = city['S_total'].values

    if mode == 'dwelling':
        C1 = 0.9
        #p_ecl = 2
        p_ecl = city['P_lighting_S'].values
        C34_ref = 0.5
        Elnat_ref = 200
    elif mode == 'tertiary':
        C1 = 0.8
        p_ecl = 5 #puissance installée W/m2
        C34_ref = 0.03
        Elnat_ref = 700.

    FfDir = coef_shading
    FfDiff = coef_shading

    A_baie = city['S_window'].values
    A_eclnat = city['S_indoor'].values

    T_l1 = 0.8 #facteur de transmission lumineuse (valeur à
    verifier) -10%

    D_fv11 = (1.8*T_l1*A_baie)/(4.5*A_eclnat*0.75)

    Drp = (I_dir_s + I_dir_n + I_dir_e + I_dir_o) / 4.
    Dfp = I_dif_surf.copy()
    Rrp = I_ref.copy()
    Eb_dir = 100 * Drp * FfDir
    Eb_dif = 100 * Dfp * FfDiff
    Eb_ref = 100 * Rrp
    Eb_eq = Eb_dif + (0.2*Eb_dir + 0.6*Eb_ref)

    D_fv11 = D_fv11[np.newaxis].T

    Eb_eq = np.mat(Eb_eq)
    E_inat = np.dot(D_fv11, Eb_eq)
    E_inat = np.asarray(E_inat)
    a1 = (C34_ref-1) / (Elnat_ref-100.)
    b1 = 1 - a1 * 100
    a2 = (0-C34_ref) / (2800.-Elnat_ref)
    b2 = C34_ref - a2 * Elnat_ref
    #C34 = np.zeros((NB_dwe,8760))

    C34 = ((E_inat <= Elnat_ref) & (E_inat >
    100)).astype(int)*(a1 * E_inat + b1) \
    + ((E_inat > Elnat_ref) & (E_inat <=
    2800)).astype(int)*(a2 * E_inat + b2)

    C34[E_inat <= 100] = 1
    C34[E_inat > 2800] = 0 #La valeur de la rt2005 est 2800
    if mode == 'dwelling':
        C34[occ.T == 0] = 0
        C34[rest_resu.T > 0] = 0
    else:
        C34 = C34*occ.T
    #c_ecl = p_ecl * (C1 *np.tile(b_sur, (len(occ[:,0]),1)).T
    * C34).T #consommation électrique

    c_ecl = np.multiply(C34.T, p_ecl * C1 *b_sur)
    c_ecl = np.multiply(c_ecl, city['weight'].values)
    return c_ecl.T
```

Données d'entrées:

INSEE « Emploi du temps » 2009-2010

- 20000 carnets d'activités de 24h au pas de temps 10 minutes
 - Situation professionnelle
 - Type de jours («normal», vacances, arrêt maladie...)
 - Jour de la semaine
 - Pondération
 - Activités principales et secondaires (non négligeable)
- Réduction de 111 activités à 10 par association :

Activity Name	details	Type of consumption
Digital	TV, computer, video game, DVD player, phone	Electricity
Cooking	Cooking plate, microwave, oven, cook top	Hot water, gas and electricity consumption
Meal	Action of eating	None
Rest	Nap and sleep	Lighting turn off
Personal car	Bath, shower, body care,	Hot water consumption
Other leisure	Game, homework, reading....,	None
Other housework	Hoover, iron	Electricity
Wash the dishes	Hand washing or dishwasher	Hot water and electricity
Laundry	Hand washing or with a washing machine and dryer	Hot water and electricity
Battery charge (VE)	<i>En cours ...</i>	<i>En cours ...</i>

Transformation des données EDT pour Smart-E

1. Les jours de la semaine et les weekends sont discriminés
2. Les actifs, les retraités, les chômeurs et les « autres » sont discriminés
3. Un calendrier d'activité d'une semaine est généré pour chaque occupant en concaténant des journées pseudo-aléatoires
4. Les occupants sont « additionnés » pour créer des foyers
5. Une puissance moyenne est associée à chaque activité
 - Les activités «lave-linge» et «lave-vaisselle» sont traitées séparément
6. Un ratio de gain interne, des débits d'eau, des possibilités d'action sont associés à chaque activité

Hypothèses limitantes

- Une personne peut effectuer jusqu'à deux activités en même temps
- Lorsque plusieurs personnes font la même activité, la puissance ne varie pas
- Une puissance constante est associée à chaque activité, elle peut varier d'un logement à l'autre
- Les personnes d'un même foyer n'ont pas « d'interaction entre eux »
- Pas d'évolution des comportements en fonction des saisons

Données d'entrées: INSEE recensement 2008 et 2010

Census information	Models Used
Dwelling construction year	Envelope performance, retrofit probability
Dwelling area	Inertia, heated area, heat loss, DWH sizing
Energy for heating	Type of HVAC system
Household composition (age, activities)	Appliances used, hot water system, internal heat gain
Number of occurrence in city	Weight of dwelling in city
Single or multi-housing	Heat loss surface
Localization	Weather data



Données peu précises :

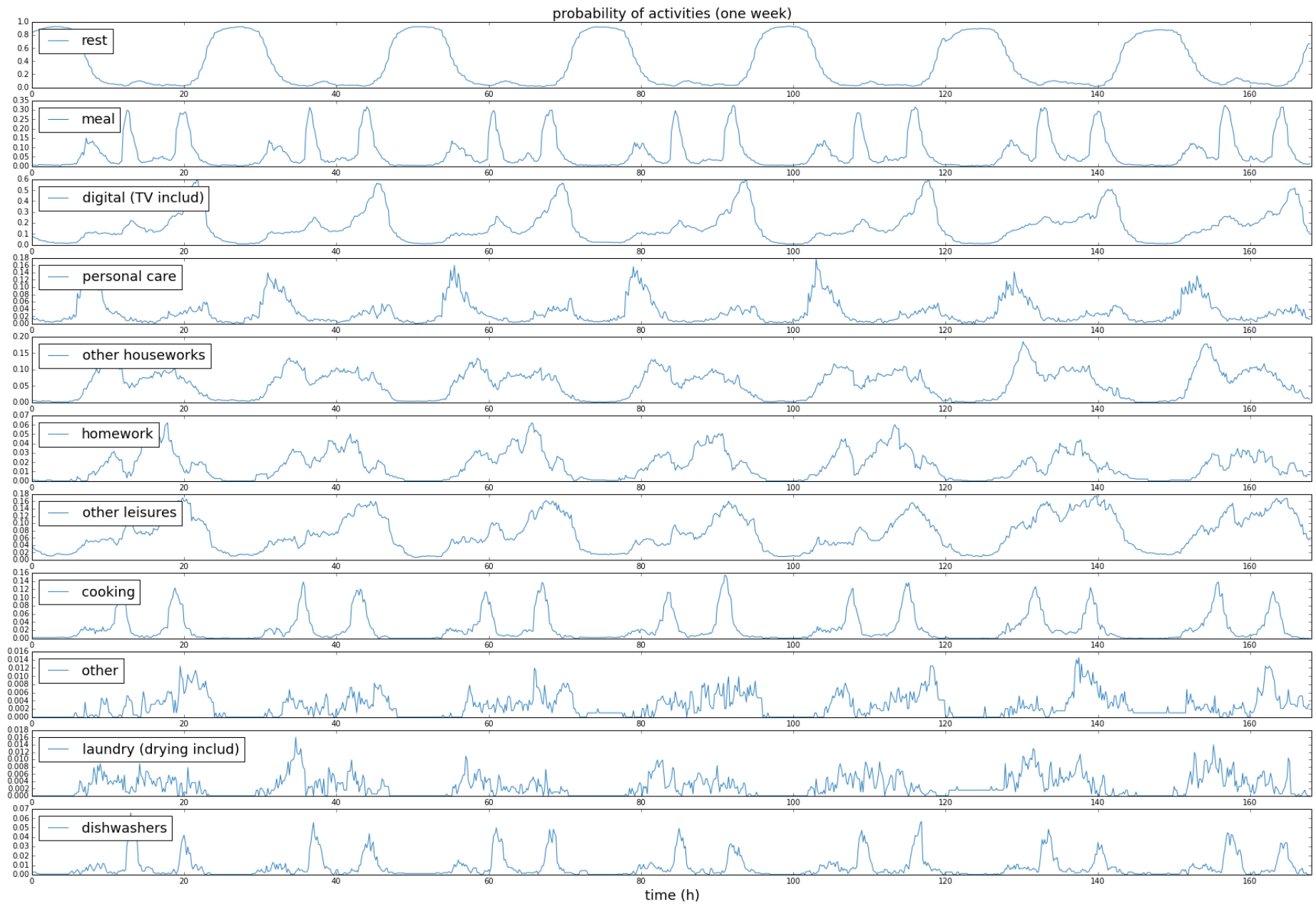
- Années de construction
- Surface

Données non connues

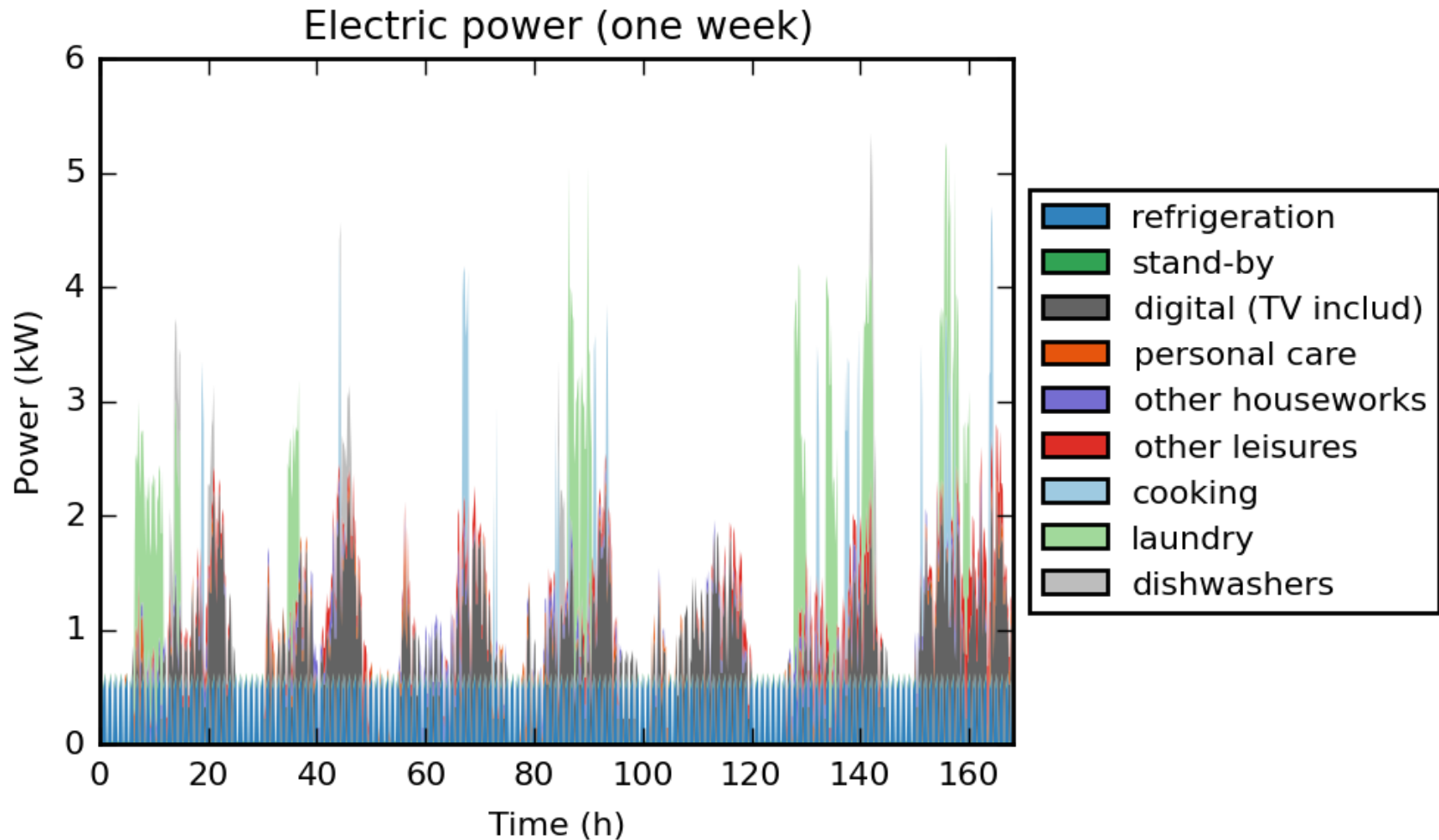
- Année et type de rénovation
- Système de production d'eau chaude sanitaire
- Système CVC

3. SIMULATION D'UN PARC RÉSIDENTIEL - EXEMPLES DE RÉSULTATS

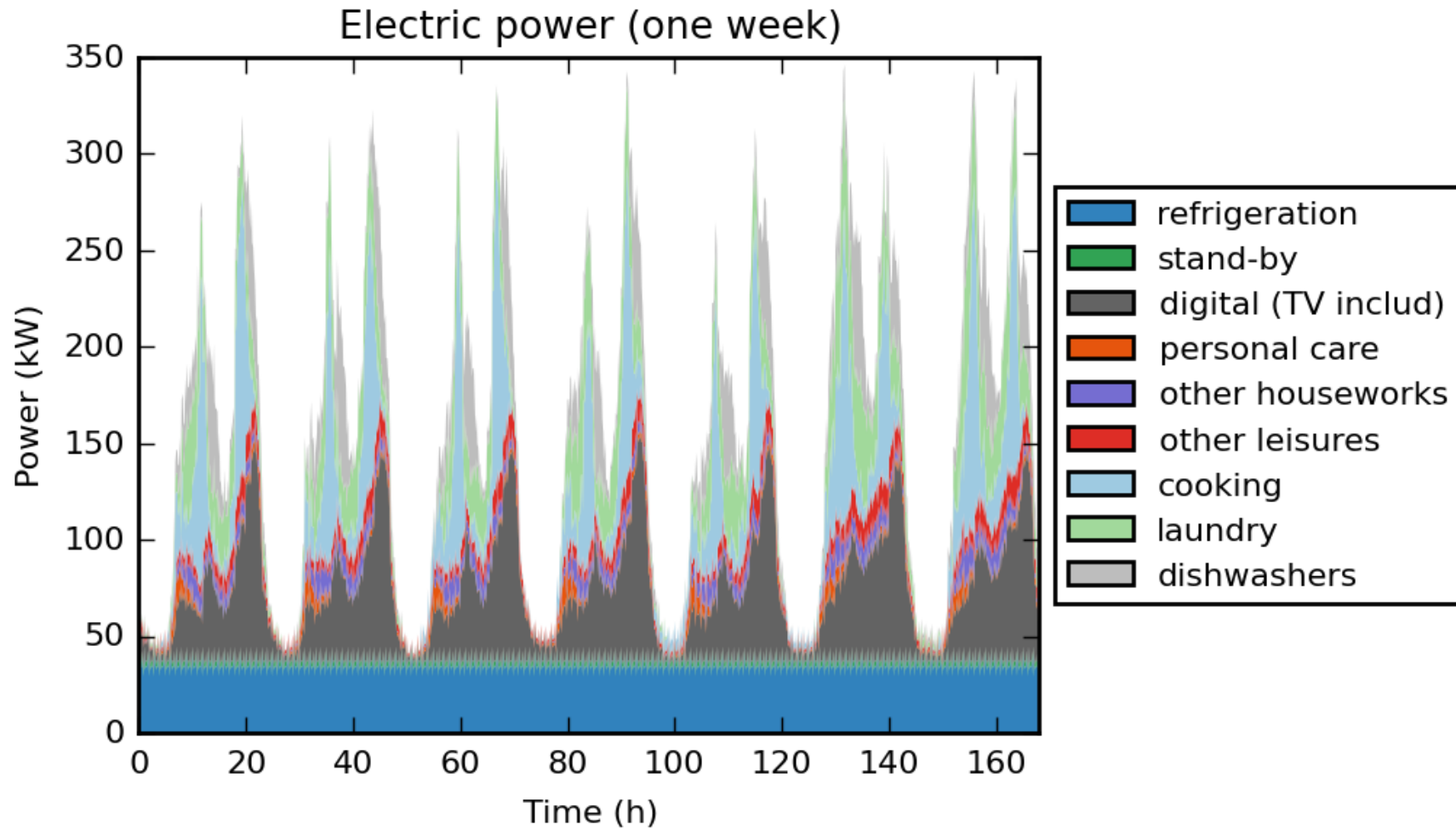
Cohérence visuelle par activité



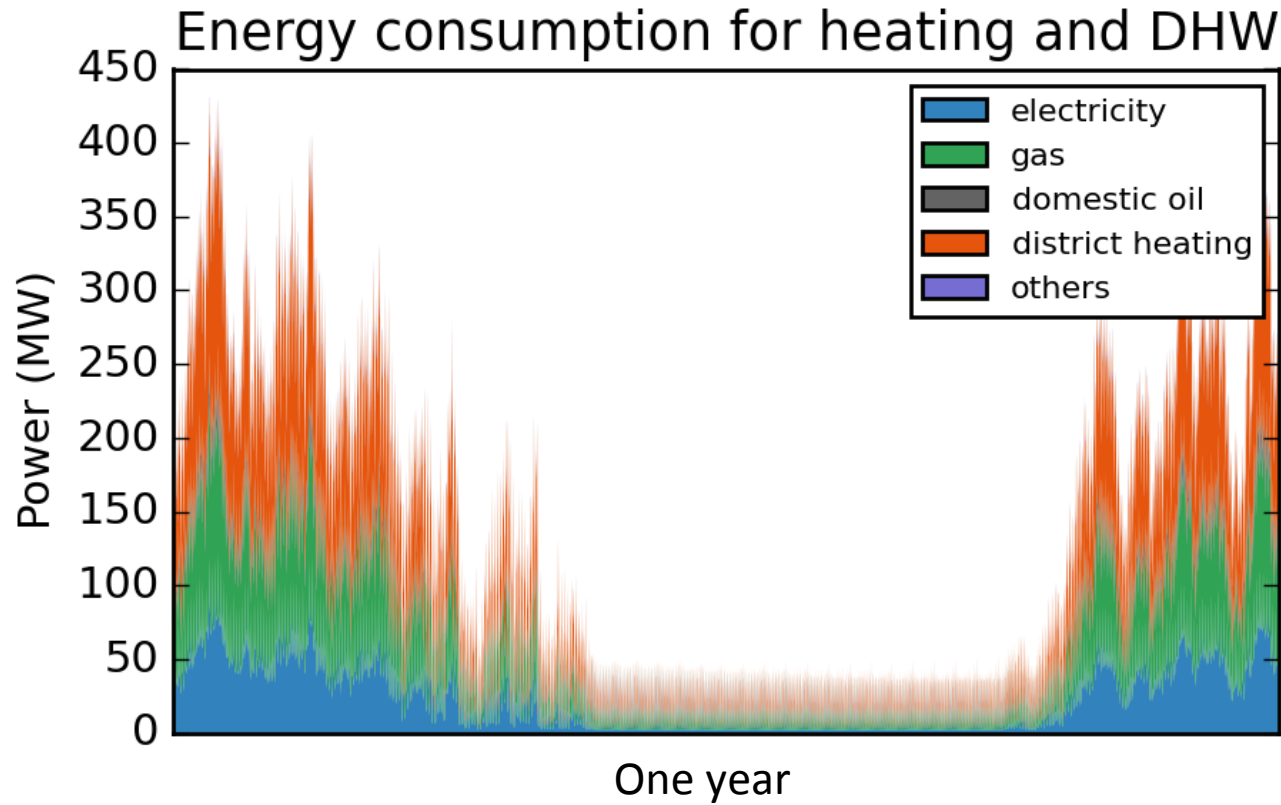
Cohérence visuelle des consommations électriques (5 logements)



Cohérence visuelle des consommations électriques (500 logements)

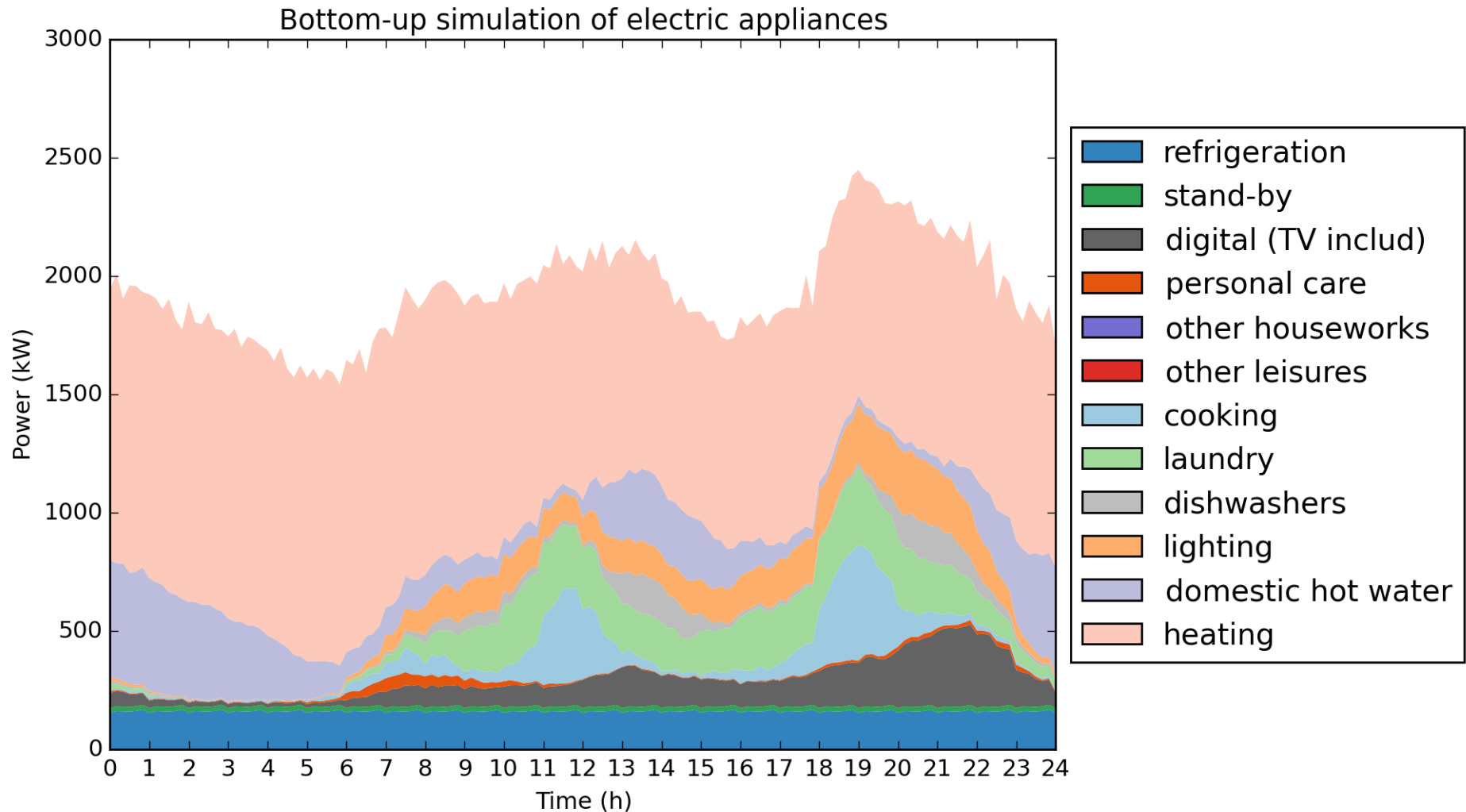


Simulation réalisée avec Smart-E : 13^{ème} arrondissement de Paris



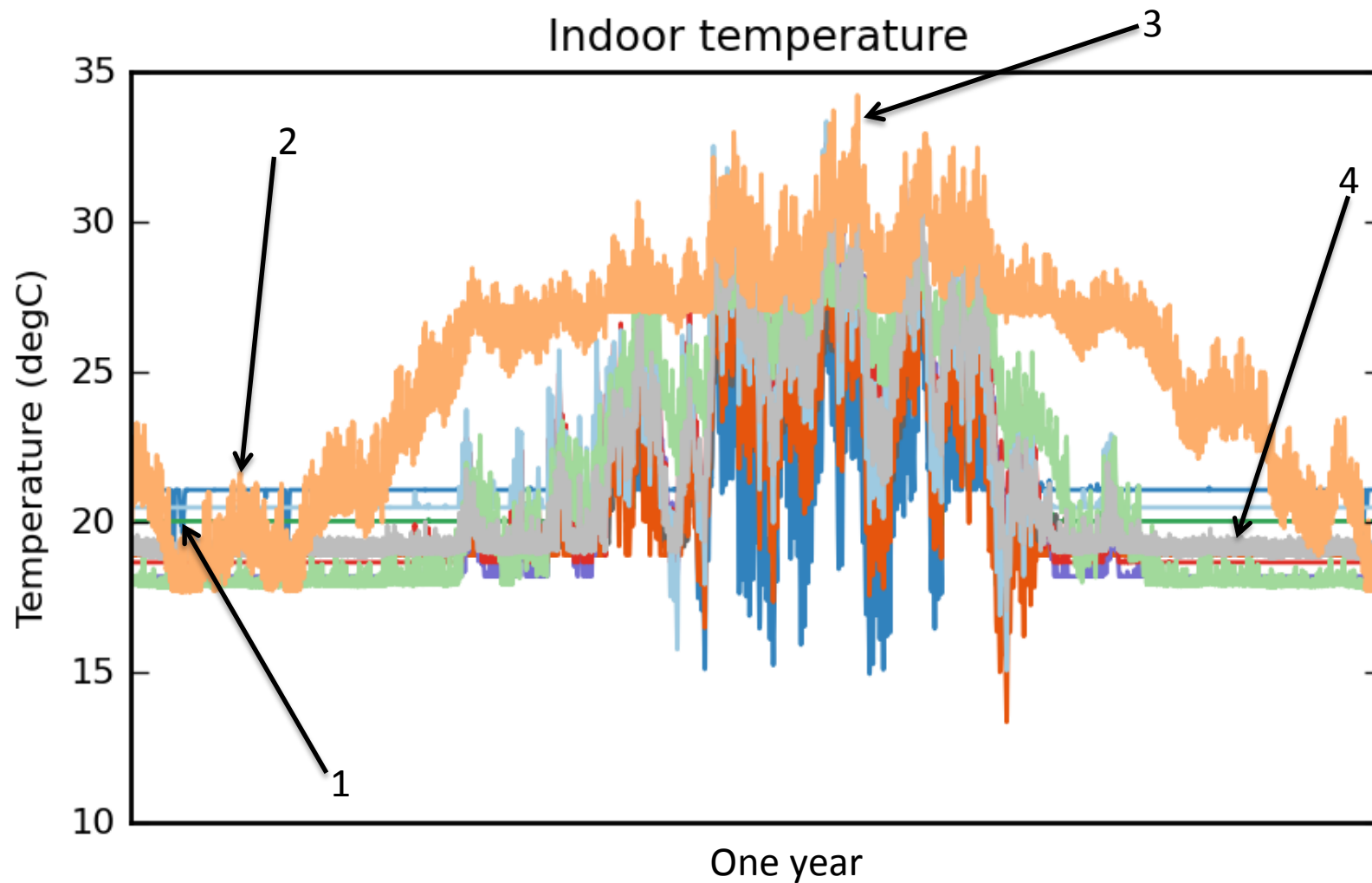
↑ 13^{ème} arrondissement (98 000 logements)

Simulation réalisée avec Smart-E : 13^{ème}



13^{ème} arrondissement (2 000 logements) : superposition des usages de l'électricité sur une journée froide en semaine, 31% des logements sont équipés d'un système de chauffage électrique

Cohérence visuelle des températures intérieures moyennes (10 logements)



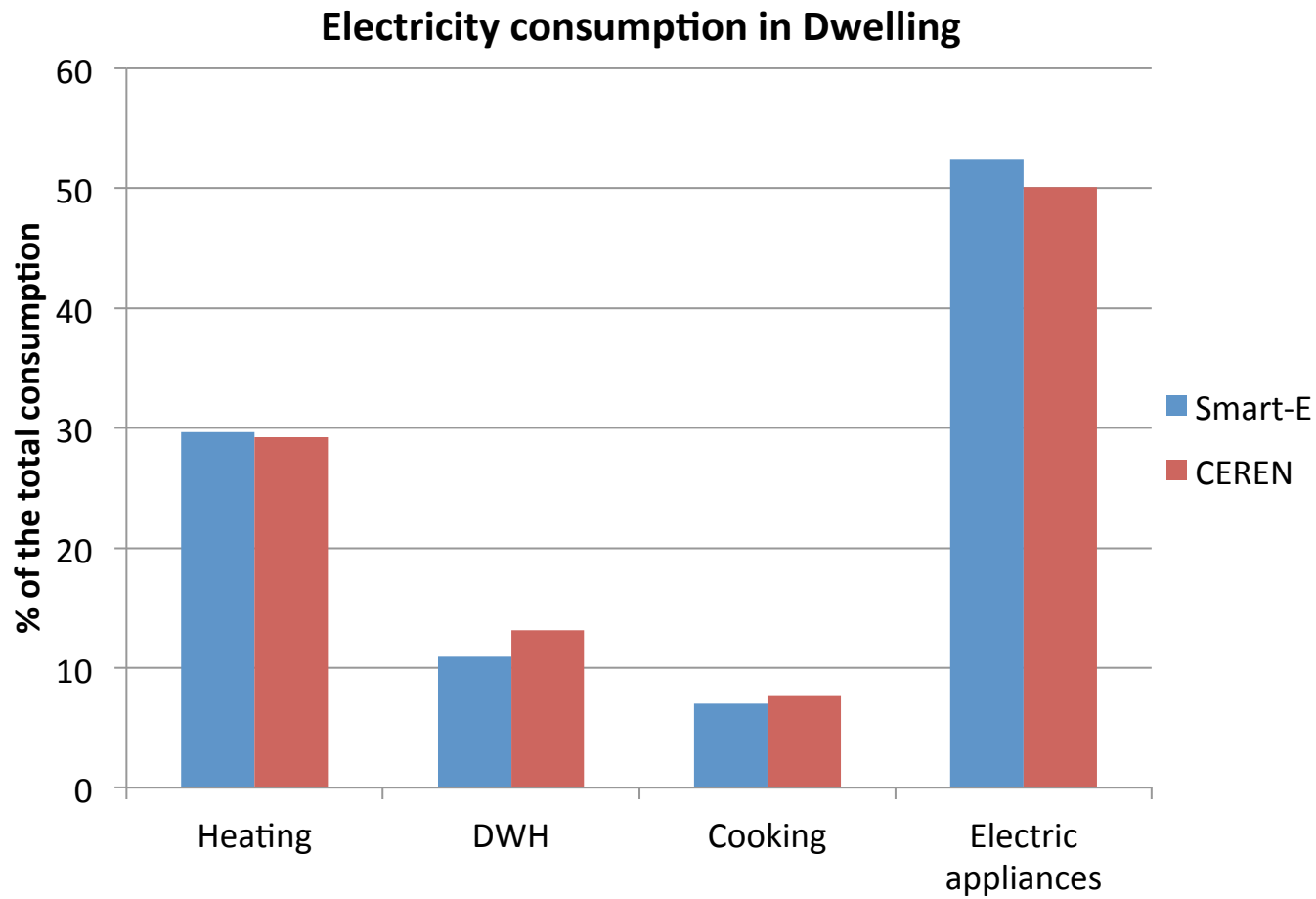
4. VALIDATION D'UN MODÈLE ÉNERGÉTIQUE DE PARC

La validation : un verrou scientifique

- Deux approches complémentaires
 - Campagne de **mesures**, plusieurs échelles possibles :
 - Territoires (énergies)
 - postes de distribution (consommation ~1h)
 - Bâtiments ou résidences (consommation ~10 min)
 - Logements (consommation ~10 min usage par usage)
 - Des méthodes statistiques
 - Méthodes Bayésiennes
 - Analyse de sensibilité
 - Utilisation du phénomène de foisonnement
 - **Comparaison d'outils**
- Choisir des territoires de références pour comparer et valider les outils de modélisation de quartier

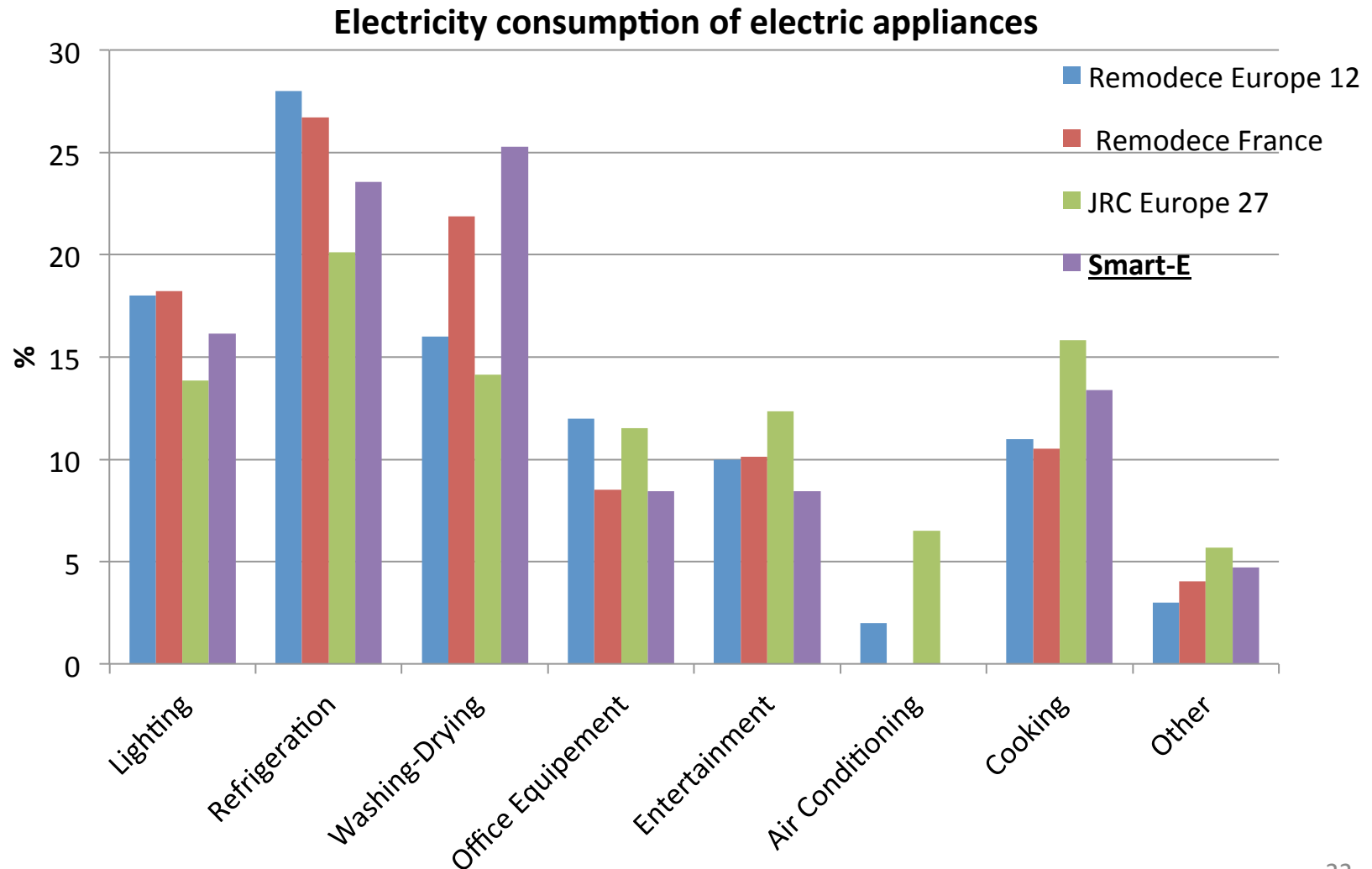
Validation en énergie annuelle :

Comparaison d'un parc de logements avec le parc français

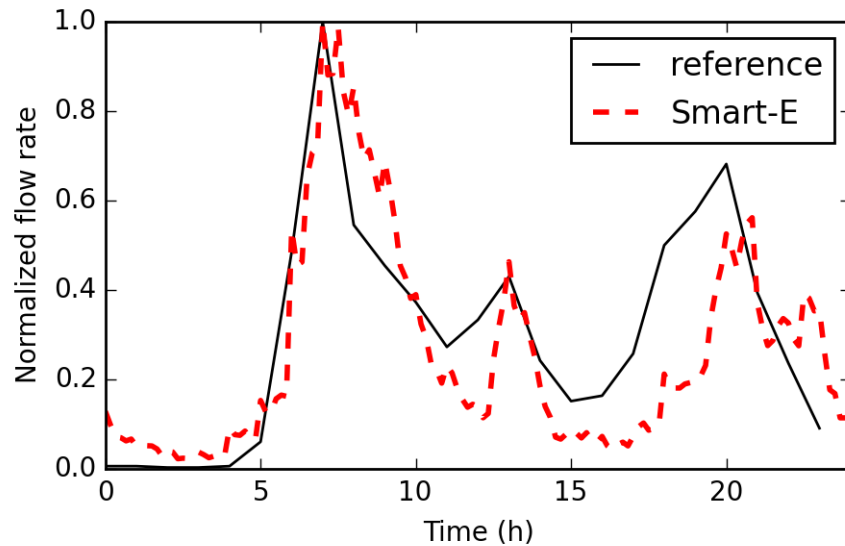


Validation en énergie annuelle :

Comparaison d'un parc de logements avec le parc français



Exemple de validation de la consommation électrique de l'ECS à l'échelle de la France



➔ Validation de consommation d'eau chaude : comparaison Smart-E et **AICVF** sur une journée en semaine

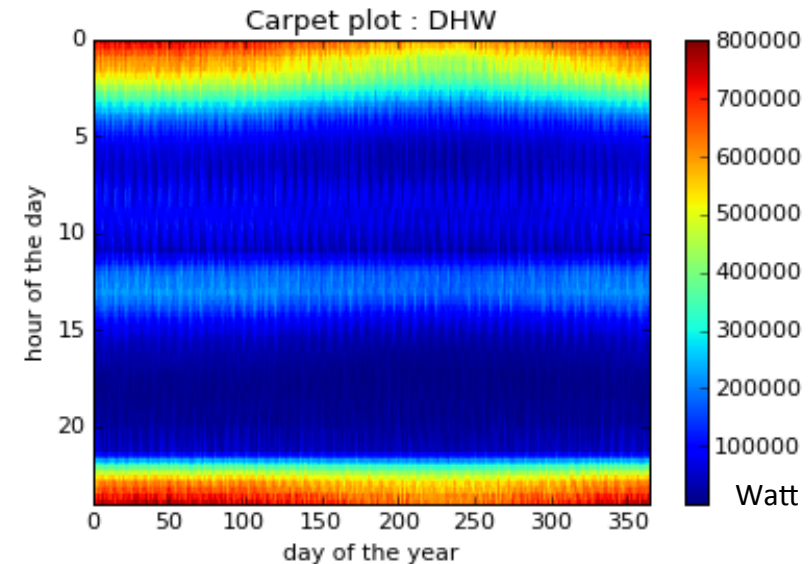
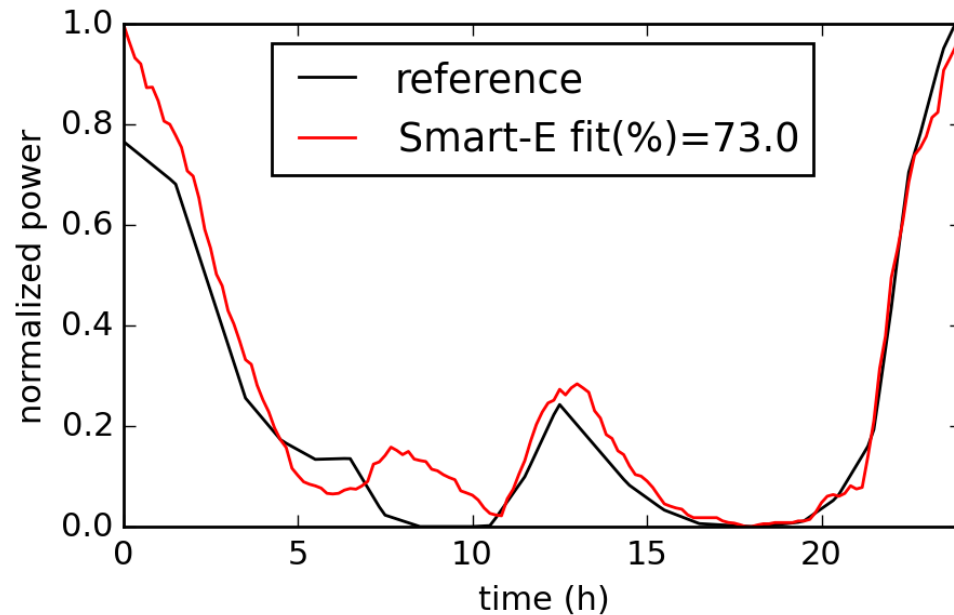
➔ Validation de la courbe de charge électrique: comparaison Smart-E et **RTE** sur une journée froide

Atelier d'aujourd'hui

Préparation de l'Atelier

- Téléchargement d'une version simplifiée de Smart-E :
 - <http://www.ces.mines-paristech.fr/Logiciels/Smart-E/>
 - Ou rechercher « Smart-E Mines » sur internet puis cliquer sur Smart-E SIMUREX en bas de page
- Téléchargement de Anaconda de Continuum Analytics ~300Mo (python, numpy, pandas, matplotlib)
- Déterminer par méthode inverse les ordres de mise en route des ballons d'eau chaude par ERDF
 - Modifier la variable « DHW_time » du tableau « city_simurex.csv »
 - Les Heure de mise en route sont comprises entre 0 et 143 (pas de temps de 10 min)
 - Utilisation d'Excel ou de Python au choix
 - Objectif : maximiser le « fitting »
 - Méthode d'optimisation : essai-erreur

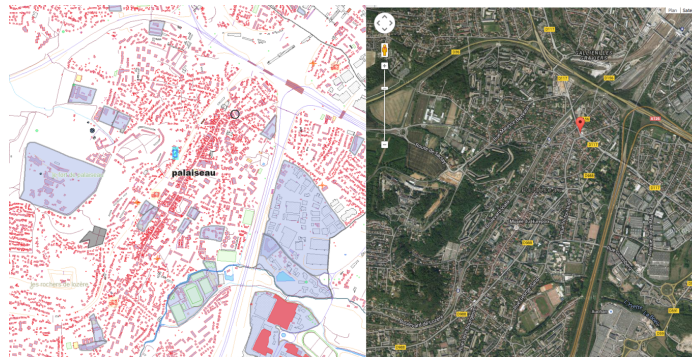
Résultats de l'Atelier



```
#solution proposée
city['DHW_control'] = (city['DHW_tank'].values == 1) * np.random.choice([0,1,2], NB_dwe, p=[0.15, 0.12, 1-0.15-0.12])
t1 = 0 #pas de temps de début et fin de commande des chauffes eau (10 min)
t2 = 24
t3 = 66
t4 = 84
t5 = 128
t6 = 144
p1 = np.linspace(1,0,t4-t3)/sum(np.linspace(1,0,t4-t3))
coef = 1.5
p2 = np.r_[np.linspace(1,0,t2-t1) , coef*np.ones(t6-t5)] / (sum(np.linspace(1,0,t2-t1)) + sum(coef*np.ones(t6-t5)))
city['DHW_time'] = \
+ (city['DHW_control'].values == 0).astype(int)*-999\
+ (city['DHW_control'].values == 1).astype(int)*np.random.choice(np.arange(t3,t4), NB_dwe, p = p1)\
+ (city['DHW_control'].values == 2).astype(int)*np.random.choice(np.r_[np.arange(t1,t2),np.arange(t5,t6)], NB_dwe, p = p2) # not controled (24%),
```

Perspectives pour Smart-E

- Validation en énergie et puissance (en cours)
 - Nombre de cycle
 - Thermo-sensibilité
 - Analyse des pointes
 - Formes des appels de puissance journalières
- Utilisation d'informations géo localisées (APUR, IGN, Google Maps)



- Réseaux et systèmes de stockage (en cours)
- Aide au pilotage d'un Smart Grid réel
- Intégration des bâtiments tertiaires et de systèmes CVC complexes
- Diffusion de Smart-E en open-source pour créer une communauté d'utilisateurs (en cours de discussion)