IZARRA GREEN-M^2 - GREEN MIDDLEWARE & METHODOLOGY

CRI / IRIT / LIP / LIUPPA / CRESTIC / APL

RELATED DOMAINS

- GreenIT researches focus on hardware efficiencies (recycling, materials, etc.)

- Some scientists focus on DataCenters & middlewares using more virtualization, load balancing technics, image migration
  - HPC, Cloud

- At a lower scale, some computer scientists focus on the use of languages and how they develop code and implement numerical services
  - Languages, good practices

EYE OPENER !

- A program is, of course, code, but also a software engineering approach.

- Hypothesis :

An eco-responsible software-engineering approach will strongly benefit applications energy consumption.

=> If an application (including interactions) is well designed (i.e. eco-responsible designed!) we can increase and optimise performances-energy consumption according to users and application needs with autonomic technics.

OBJECTIVE

- **Izarra - Green M2**
  - implement applications that are environmentally friendly with a green integrated environment including a methodology and a middleware.

INSTINCTIVELY

- Acting on both processing and data helps optimizing energy consumption.

- For **interactive mobile applications** and their evolving usages, it is extremely difficult to reach this objective in the long term.

- Requirements are opposite:
  - energy optimization and an optimal use of resources suppose a dynamic distribution of data and processes.
  - performance and good usability for end-users suppose service delivery and interaction continuity.

PRACTICAL OBJECTIVE

1. Proposing technical solution (middleware) able to migrate/replace/duplicate software component/services from/to hosts (interactive devices, cloud, IoT, etc.) as well as interactions objects (widgets, interaction modality, etc.)
   - Software architecture, middleware

2. Decision algorithms in order to decide on-the-fly reconfiguration (migration/replacement/duplication/suppression)
   - Dynamic deployment, autonomic computing (functional aspects)

3. Decision algorithm to migrate, duplicate, delete [mobile] data in order to optimize their consumption/transfer.
   - Mobile data management Autonomic computing (data aspects)

BUT…

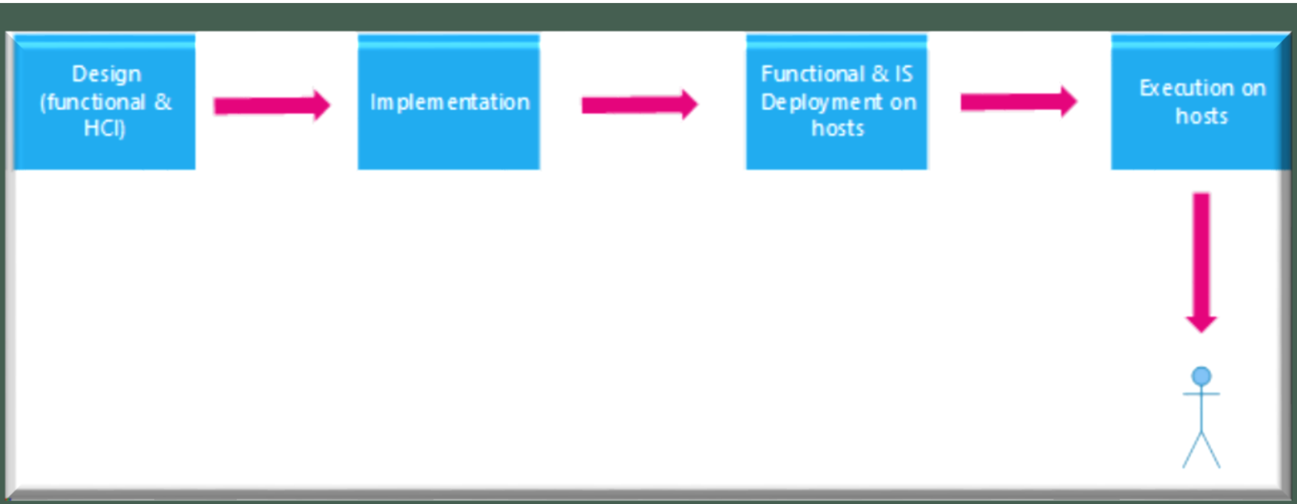
- Such middleware and algorithms would be inefficient if they are not followed by some guidance for software developers during design process

- Software engineering approach
  - Propose a green oriented design method dedicated to mobile applications, guiding software developers through practical rules, best practices and DSL integration.

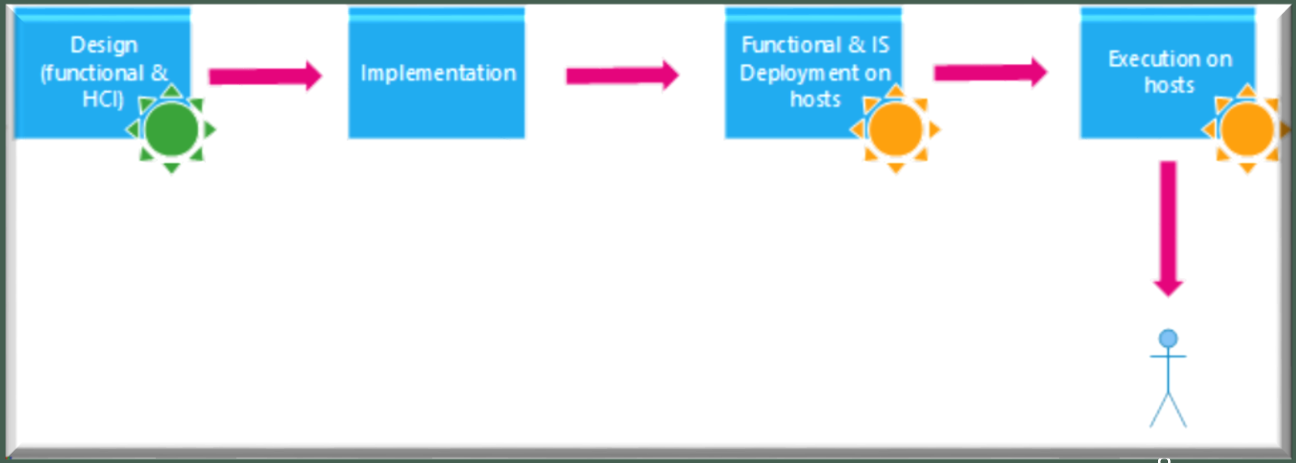
Assertion: a global approach acting at all software (software engineering -> implementation -> execution) is the only solution to really provide eco-responsible approaches acting at all levels, and the only one able to produce eco-aware applications

The scientific objective is to propose a design method, a green middleware and dedicated green oriented autonomic algorithms for eco-responsible mobile applications.

SOFTWARE ENGINEERING APPROACH


Design (functional & HCI)
Implementation
Functional & IS Deployment on hosts
Execution on hosts

- Green -> Offline
- Orange -> Run-time


Design (functional & HCI)
Implementation
Functional & IS Deployment on hosts
Execution on hosts

ORGANIZATION

- LIUPPA / T2I – University of Pau: **Philippe ROOSE, Marc DALMAU, Yon DOURISBOURE, Pierre DIBON**
- IRIT / SEPIA– University of Toulouse: **Jean-Marc PIERSON, Georges DA COSTA, Patricia STOLF, Amal SAYAH**
- LIP / Inria AVALON – University of Lyon: **Laurent LEFEVRE, Jean-Patrick GELAS**
- CRI – University of Paris 1 - Panthéon Sorbonne: **Manuele KIRSCH PINHEIRO, Carine SOUVEYET**
- CRESTIC - University of Reims Champagne Ardennes: **Luiz Angelo STEFFENEL**
- APL – **Caroline VATEAU**

WPS

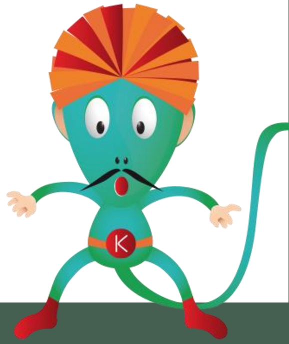
1. State of the art on Design Method Identification and Techniques related to Green IT
   - Develop a best practice catalogue and action points for all levels (Components, components Assembly and components and data Deployment).
2. Autonomic Green Oriented Deployment Algorithms & Kalimucho Integration
   - Propose an autonomic algorithm to (re-)deploy   components on hosts according to green primitives/preoccupations
3. Adaptation rules
   - Propose tools for users to design eco-responsible applications and help them to express some functional aspects.
4. Eco-responsible Design Method
   - Development of Model Driven approaches integrating eco-responsible KPIs.
5. Scenarios, Prototypes & Evaluations
   - Identifying scenarios and deploying prototypes on real use cases

EXISTING STUFFS

- Kalimucho – Middleware

- Kaligreen V1; V2; – Algorithm for sustainability
  - 1st release (done)
  - 2nd release (tomorrow !)

LA BASE DE TOUT…LE MIDDLEWARE - KALIMUCHO



Plateforme (à service) pour applications pervasives à base de composants logiciels (VIDEO)

- **Applications Dynamiques [re-]déploiement sur périphériques mobiles** (smartphones, tablet, PC, etc.).
- **Reconfigurations à chaud** : ie. Reconfiguration sans stopper l'application
  - Quelque soit la raison
  - En fonctions du contexte : fonctionnels, énergétiques, matériel, utilisateur.
  - Points d'adaptations possibles en général : paramètres, fonctions, code/contraintes, objets, composants, assemblages, etc.)
- **Transfert d'informations entre composants logiciels**
  - Avec la gestion automatique de passerelles (Ethernet, Wifi, 3G)

KALIMUCHO

- Permet également de…

- **Réaliser des installations instantanées et temporaires** (short-lived Installation/Deployment) sur des périphériques.
  - Lorsque l’application est fermée, les composants déployés sont détruits (ou gardés en cache).

- **Accéder à des applications non résidentes**

- **Installations et déploiements ad’hoc/contextualisé selon les besoins du moment**
  - Sans passer par une opération guidée par l’utilisateur
  - Sans « [android-]market » ou « any app-store »

KALIMUCHO - DSL

- Commandes de création, suppression, migration, connexion, déconnexion et duplication des flux de sortie des composants/connecteurs.

CreateComponent nomc classe [entrée1 entrée2 ...] [sortie1 sortie2 ...]

Les listes d'entrée et/ou de sortie peuvent être vides ([null])

Une entrée ou une sortie peut être marquée "not_used" pour être utilisée plus tard

RemoveComponent nomc

SendComponent nomc vers

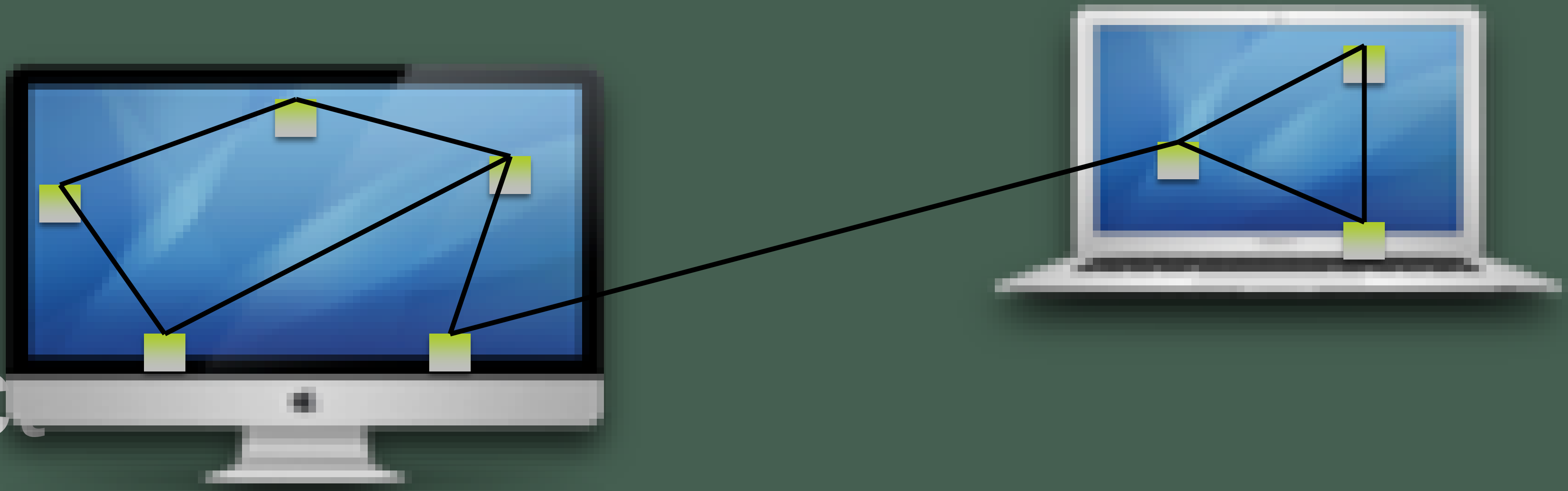
DisconnectInputComponent nomc numéro_d_entrée

DisconnectOutputComponent nomc numéro_de_sortie

ReconnectInputComponent nomc numéro_d_entrée nouvelle_entrée

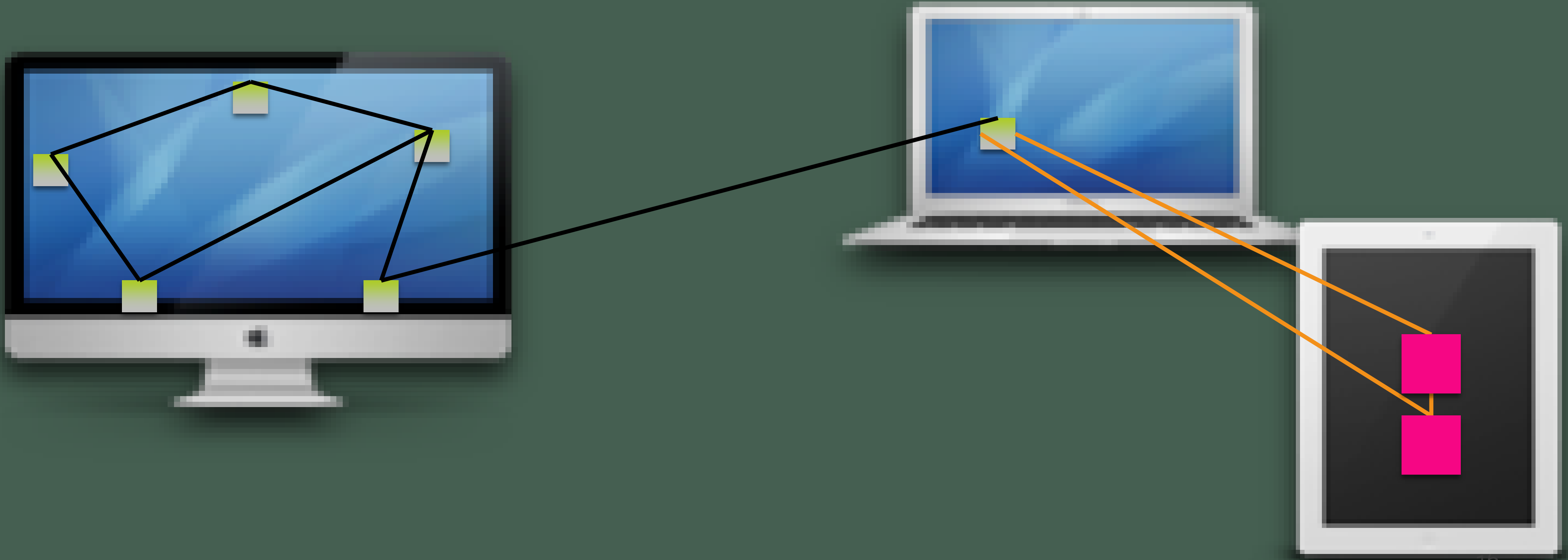
DuplicateOutPutComponent nomc numéro_de_sortie nouvelle_sortie

KALIMUCHO – ADAPTATION STRUCTURELLE

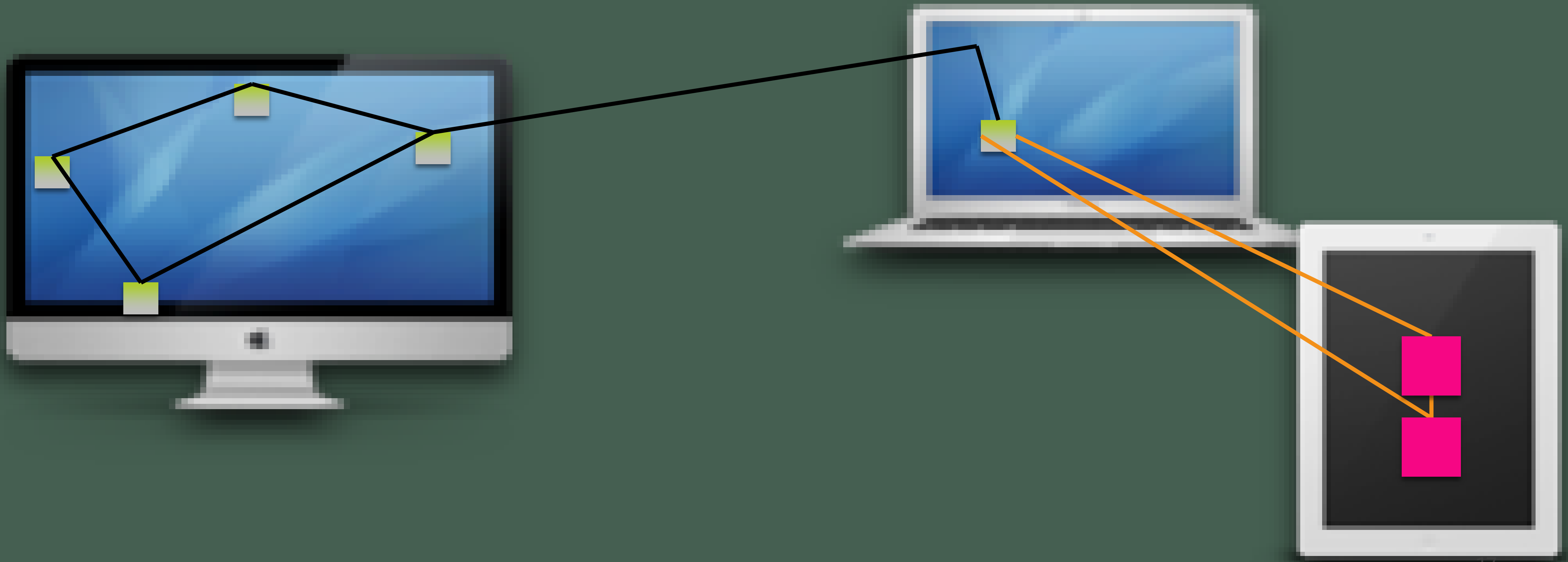


Ac
On-
The-

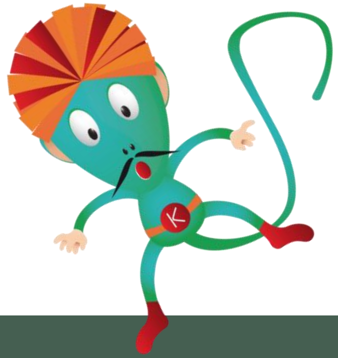
KALIMUCHO – ADAPTATION STRUCTURELLE



KALIMUCHO – ADAPTATION STRUCTURELLE



KALIMUCHO – EN CHIFFRES



  - PC (JAR) // Android(APK) < 1Mo

- **Temps d’exécution de commandes (sur Android Nexus One)**
  - Création composant: 2 à 20 ms
  - Suppression composant : 15 ms (à partir de la fin de la méthode stop)
  - Création d’un connecteur interne : 3 à 15 ms
  - Création d’un connecteur distribué: 10 à 100 ms
  - Suppression d’un connecteur interne: 3 à 15 ms
  - Suppression d’un connecteur distribué: 3 à 25 ms
  - Déconnexion/Reconnexion d’une entrée: 2 à 7 ms
  - Duplication d’une sortie: 2 à 7 ms

- **3 brevets, 1 marque déposée, Présentations CES Las Vegas, etc.**
- **www.kalimucho.com**

Mesures de Kalimucho	**PC**	**Android**
Taille	827 Ko	1032 Ko
Nombre de lignes de code	17 000	20 000
Nombre de classes	178	262
Nombre de threads lancés au démarrage	20	21
Nombre de méthodes ou de blocs "synchronized"	279	244
Nombre d'opérations "wait" et "notify"	87	72

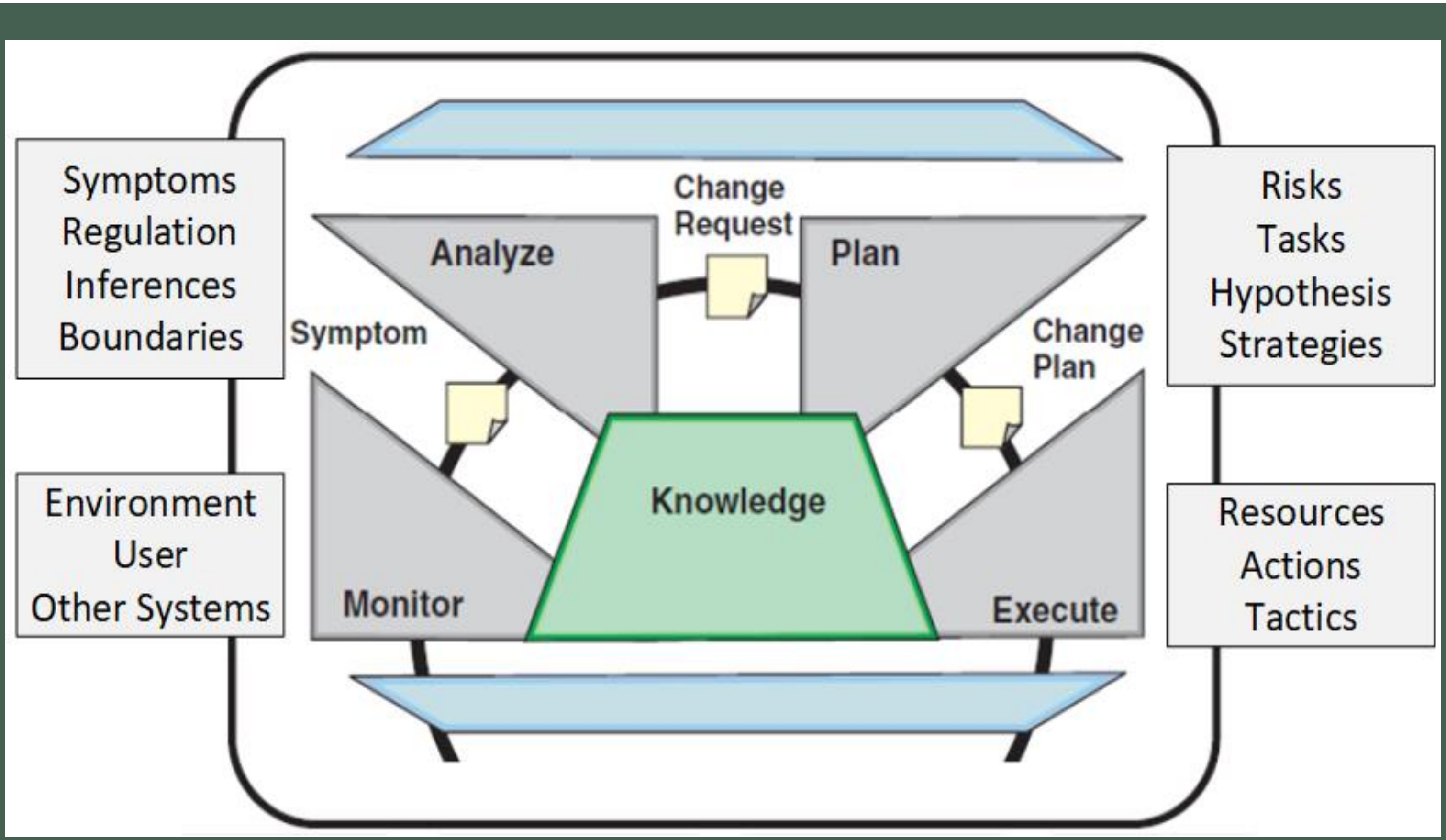
ADDITIONAL AUTONOMIC PART- KALIGREEN

- KaliGreen = Algo décentralisé permettant l’échange de services piloté par l’énergie

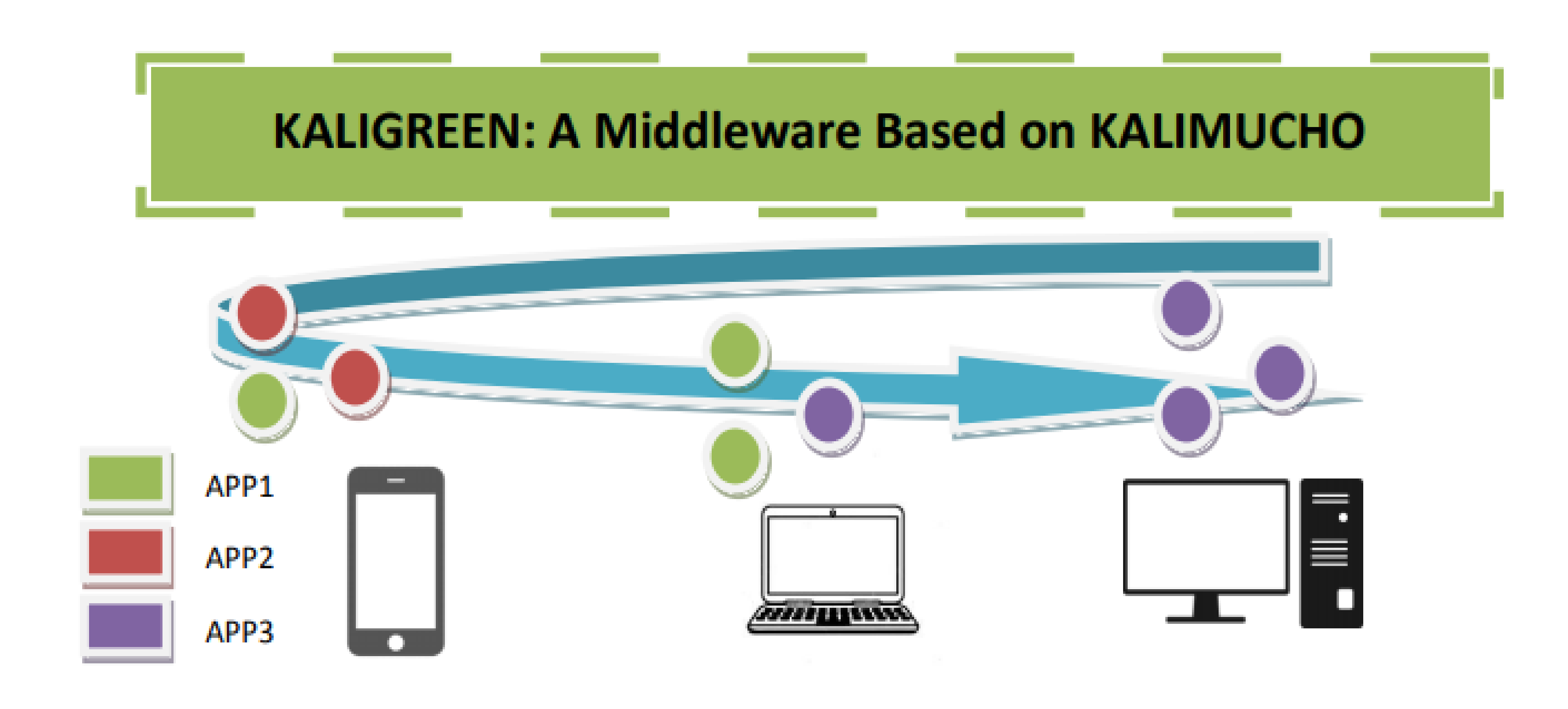
- Version light de l’algo
  - Identifier les microservices qui consomment le plus d’énergie.
  - Extraire les meta-données (CPU, taille, bande passante utilisée, etc.)
  - Ajouter dans un verteur de données
  - Envoyer le vecteur aux périphériques connectés qui évalueront la possibilité d’héberger le service
    - VectorID
    - ID du périph qui l’a produit
    - Moyenne puissance CPU nécessaire
    - Moyenne RAM nécessaire
    - Moyenne stockage requis
    - Résolution de l’écran (si necessaire) + temps moyen usage

  - Déplacer le microservice sur le meilleur périphérique hôte candidat

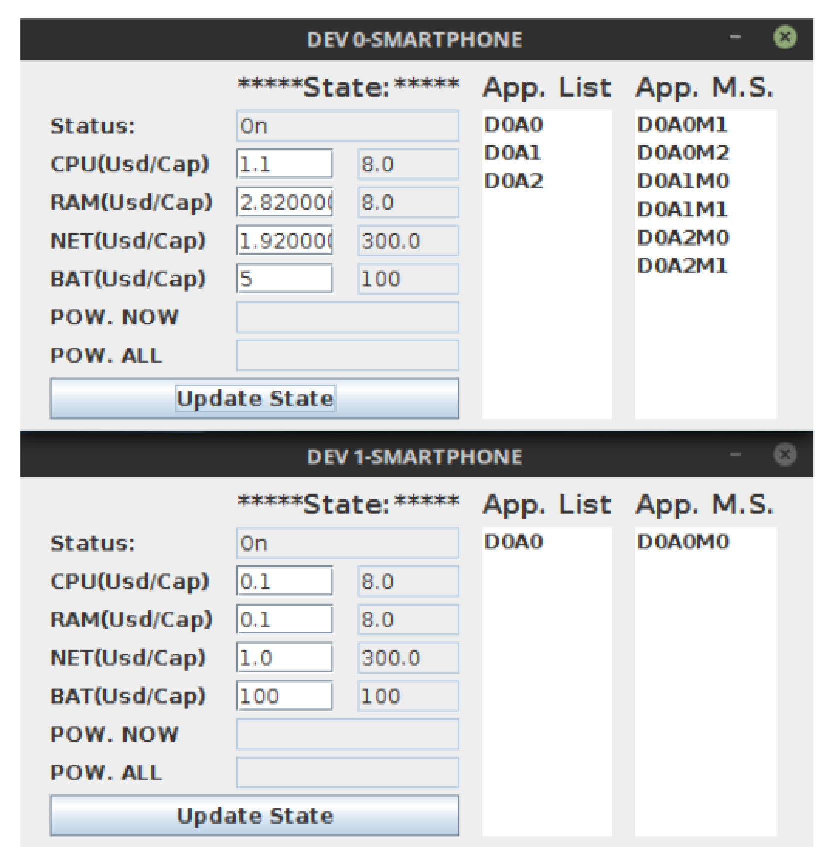
MAPE-K


Symptoms
Regulation
Inferences
Boundaries
Environment
User
Other Systems
Analyze
Change
Request
Plan
Symptom
Change
Plan
Knowledge
Monitor
Execute
Risks
Tasks
Hypothesis
Strategies
Resources
Actions
Tactics

KALIGREEN



KALIGREEN : SIMULATEUR

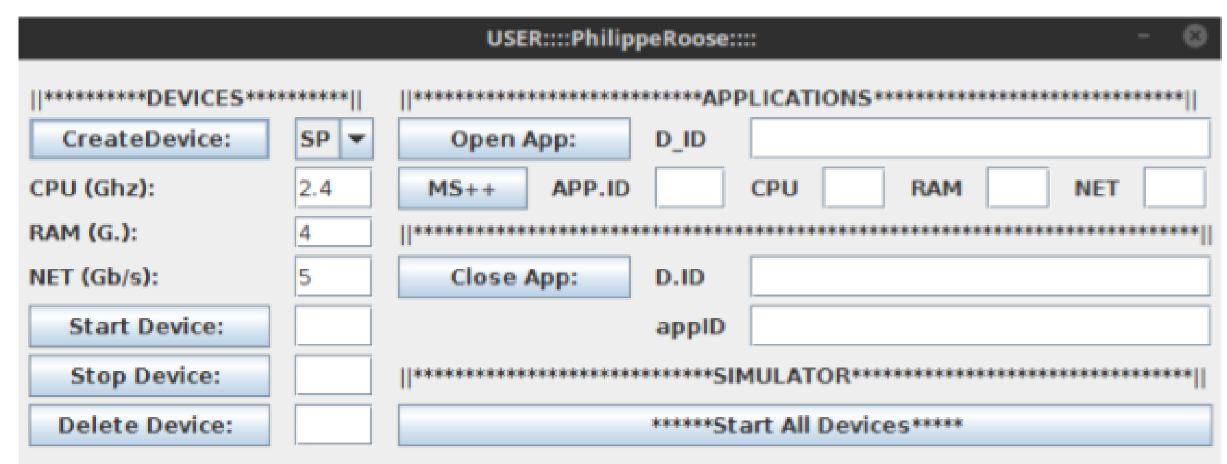


DEV 0-SMARTPHONE

	*****State:*****		App. List	App. M.S.
Status:	On		D0A0	D0A0M1
CPU(Usd/Cap)	1.1	8.0	D0A1	D0A0M2
RAM(Usd/Cap)	2.820000	8.0	D0A2	D0A1M0
NET(Usd/Cap)	1.920000	300.0		D0A1M1
BAT(Usd/Cap)	5	100		D0A2M0
POW. NOW				D0A2M1
POW. ALL				

DEV 1-SMARTPHONE

	*****State:*****		App. List	App. M.S.
Status:	On		D0A0	D0A0M0
CPU(Usd/Cap)	0.1	8.0		
RAM(Usd/Cap)	0.1	8.0		
NET(Usd/Cap)	1.0	300.0		
BAT(Usd/Cap)	100	100		
POW. NOW				
POW. ALL				



USER::::PhilippeRoose::::

||**********DEVICES**********||

||****************************APPLICATIONS******************************||

D_ID

CPU (Ghz): 2.4

APP.ID

CPU

RAM

NET

RAM (G.): 4

||**||

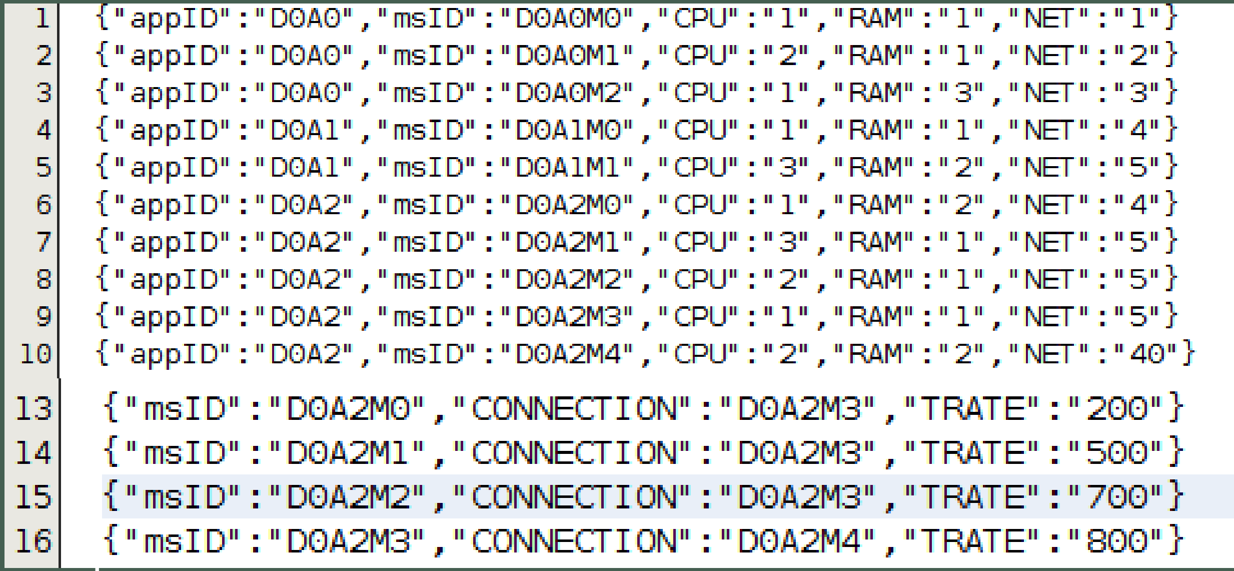
NET (Gb/s): 5

D.ID

appID

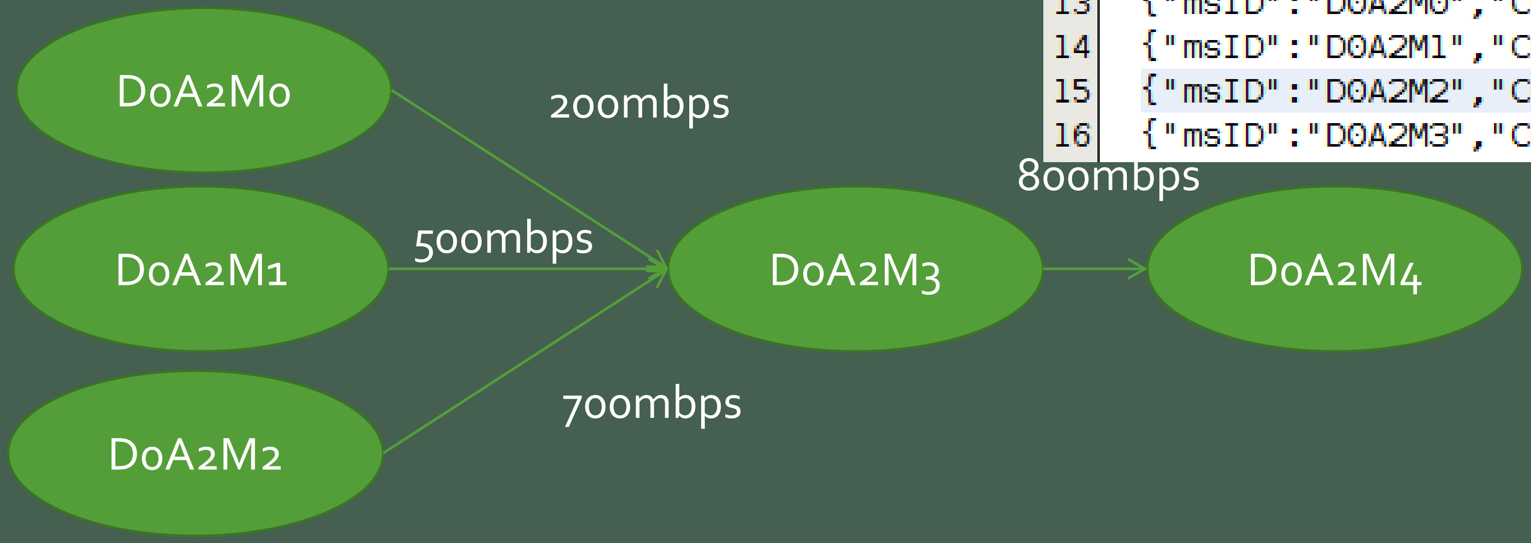
||****************************SIMULATOR*********************************||

APPS DEPLOYMENT DESCRIPTION

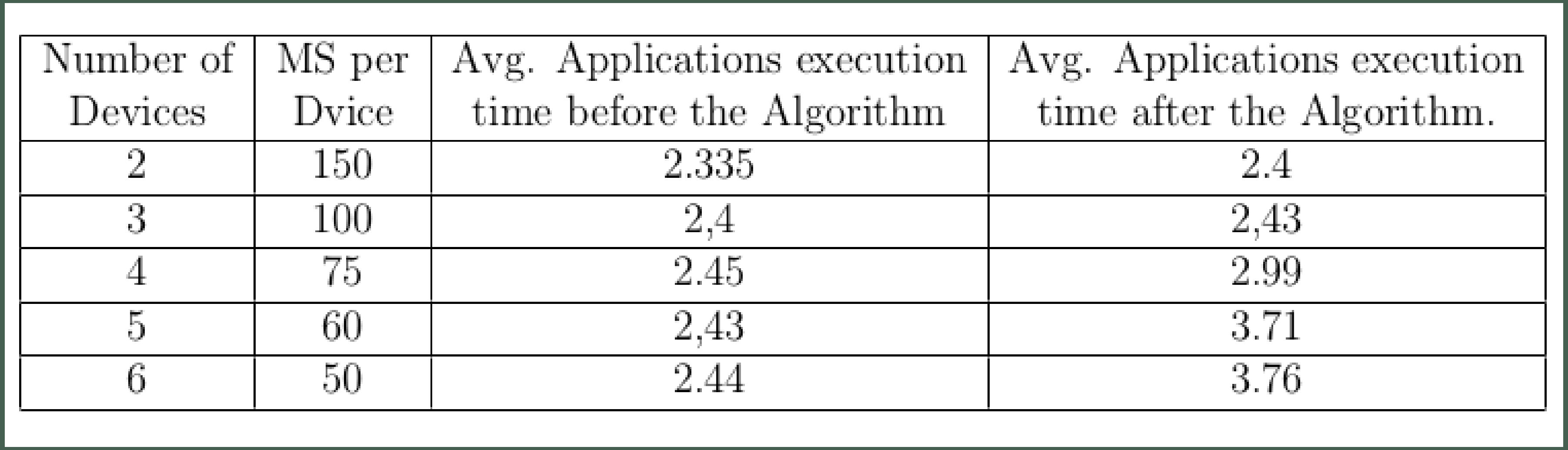


```
{"appID":"D0A0","msID":"D0A0M0","CPU":"1","RAM":"1","NET":"1"}
{"appID":"D0A0","msID":"D0A0M1","CPU":"2","RAM":"1","NET":"2"}
{"appID":"D0A0","msID":"D0A0M2","CPU":"1","RAM":"3","NET":"3"}
{"appID":"D0A1","msID":"D0A1M0","CPU":"1","RAM":"1","NET":"4"}
{"appID":"D0A1","msID":"D0A1M1","CPU":"3","RAM":"2","NET":"5"}
{"appID":"D0A2","msID":"D0A2M0","CPU":"1","RAM":"2","NET":"4"}
{"appID":"D0A2","msID":"D0A2M1","CPU":"3","RAM":"1","NET":"5"}
{"appID":"D0A2","msID":"D0A2M2","CPU":"2","RAM":"1","NET":"5"}
{"appID":"D0A2","msID":"D0A2M3","CPU":"1","RAM":"1","NET":"5"}
{"appID":"D0A2","msID":"D0A2M4","CPU":"2","RAM":"2","NET":"40"}
```

```
{"msID":"D0A2M0","CONNECTION":"D0A2M3","TRATE":"200"}
{"msID":"D0A2M1","CONNECTION":"D0A2M3","TRATE":"500"}
{"msID":"D0A2M2","CONNECTION":"D0A2M3","TRATE":"700"}
{"msID":"D0A2M3","CONNECTION":"D0A2M4","TRATE":"800"}
```



MEASURES



Number of Devices	MS per Dvice	Avg. Applications execution time before the Algorithm	Avg. Applications execution time after the Algorithm.
2	150	2.335	2.4
3	100	2,4	2,43
4	75	2.45	2.99
5	60	2,43	3.71
6	50	2.44	3.76

BACK TO IZARRA - RISKS

- The composition of the consortium and of the project scope.
  - Whereas most of works in the domain of greenIT focus on specific tasks (grid, data centers, hardware, OS, etc.), we have a global cross domain approach.
  - Each specialist has its own metrics, own preoccupations.

CURRENT

- ANR Step 1 : OK
- ANR Rebuttal phase : OK
- ANR final response : waiting !

CONCLUSION

IZARRA
Liqueur-Licor-Liquore-Likör-Likeur
IZARRA
EDARI MAITE PREGABEA
LA MARQUE DU PAYS BASQUE
40% vol.
e 700ml
Produit de France/Product of France