

# Soutenance de stage

"Analyse des données en IoT"

- ▼ **Entreprise d'accueil:** Maya Technologies
- ▼ **Durée:** 6 mois
- ▼ **Réalisé par:** GANZA Mykhailo
- ▼ **Responsable:** Mme LECOMTE Christelle
- ▼ **Encadrant:** M NACABAL François



# Introduction

- ▼ Maya Technologies
- ▼ IoT
- ▼ MayaNet

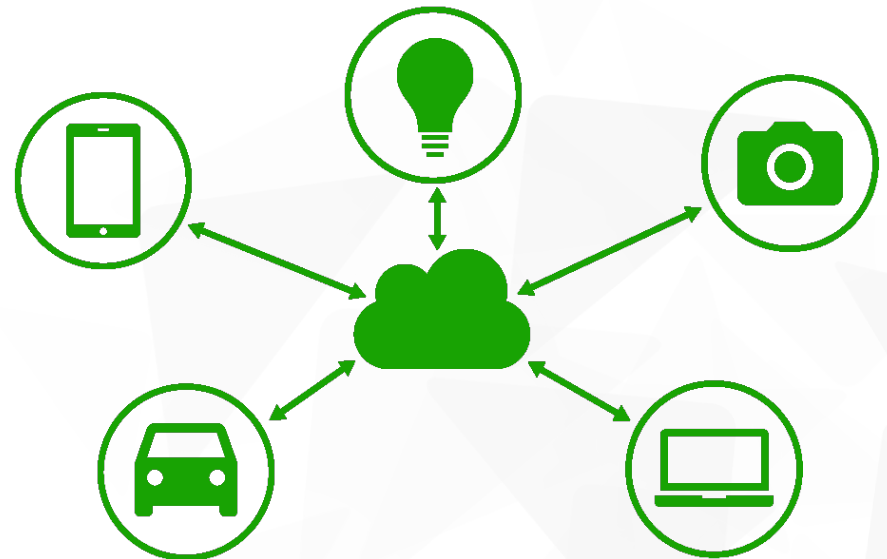
# Introduction: Maya Technologies

- ▼ Fondée en 2007 par Philippe Mattia à Grenoble
- ▼ Domaine: IoT, systèmes embarqués
- ▼ Spécialités: développement software, design hardware
- ▼ Industries: aéronautique, multimédia, santé
- ▼ Villes: Grenoble, Valence, Paris, Toulouse



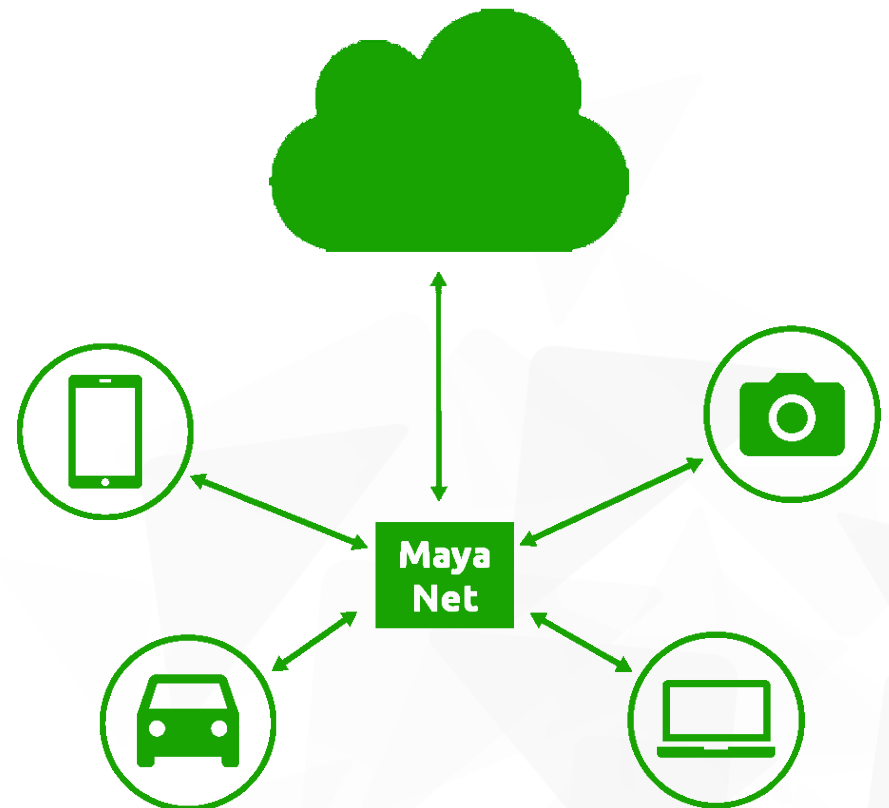
# Introduction: IoT

- ▼ **IoT** (“Internet Of Things”), les objets connectés, est une extension d'internet vers le monde physique grâce aux systèmes embarqués
- ▼ **20.8** milliards des objets connectés prévu pour 2020
- ▼ Utilisation:
  - ▼ médecine (“santé connecté”)
  - ▼ domotique
  - ▼ capteurs environnementaux
  - ▼ etc



# Introduction: MayaNet

- ▼ Une solution IoT de Maya Technologies pour la **collecte, traitement et analyse des données** provenant de différents types d'objets connectés
- ▼ Un logiciel de gestion d'un réseau des systèmes embarqués
- ▼ Protocoles supportées:  
GSM, Wi-Fi, KNX, USB, etc

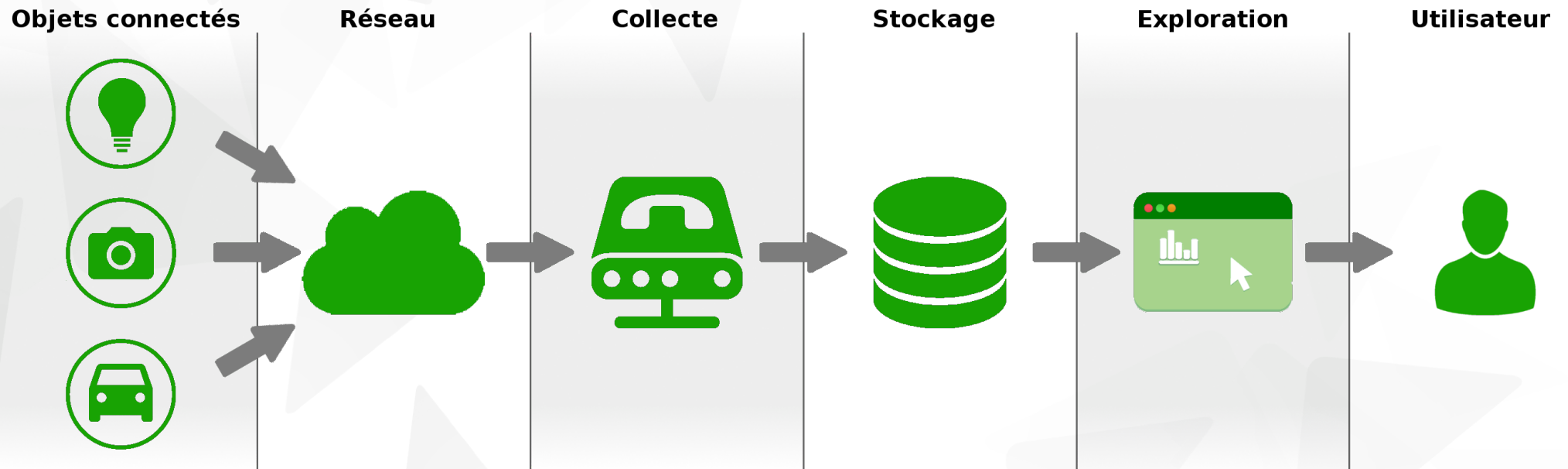


# Objectif initial

# Objectif initial

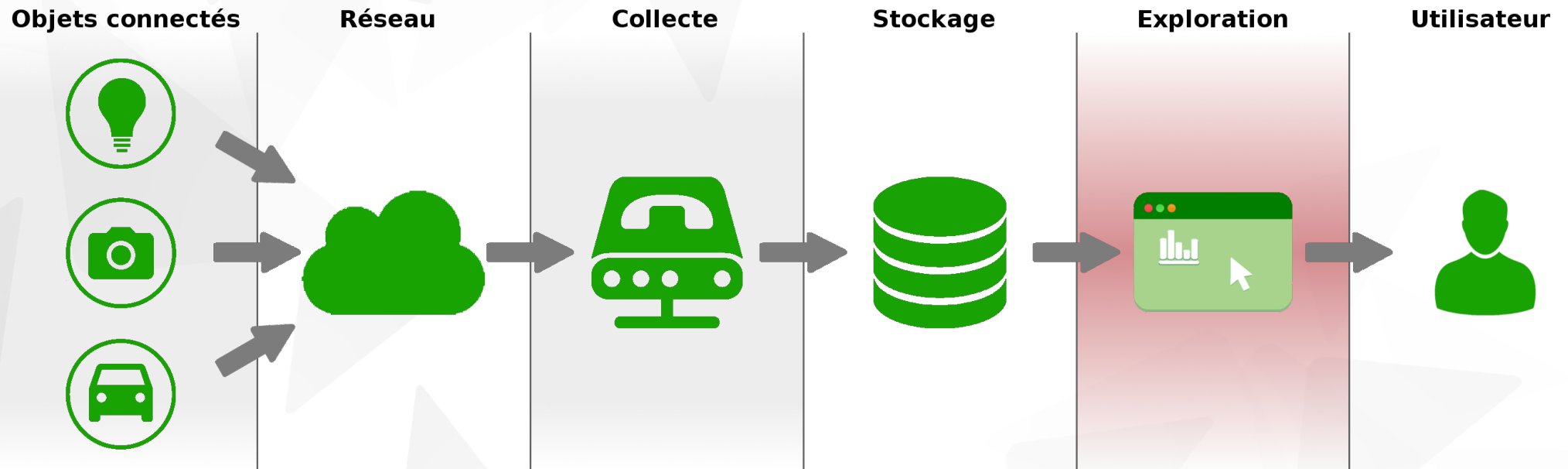
*« Concevoir un outil de **l'exploration** des **données** et les **relations** entre les données, ayant une nature différente provenant de sources différentes d'objets connectés»*

# Objectif initial (2)





# Objectif initial (2)



# Étude

- ▼ Séries temporelles
- ▼ Exploration des données
- ▼ DC.js
- ▼ Objectif concret

# Étude: séries temporelles

- ▼ Une **série temporelle** (eng: *timeseries*) est une **suite des valeurs numériques** évoluant suivant le temps
- ▼ Composé d'un ensemble des **pairs clef-valeur**, avec un clef spécifique d'horodatage (eng: *timestamp*)
- ▼ **Sources possibles**: bases des données (ex: *MySQL*, *MongoDB*), fichiers (ex: *JSON*, *CSV*), etc

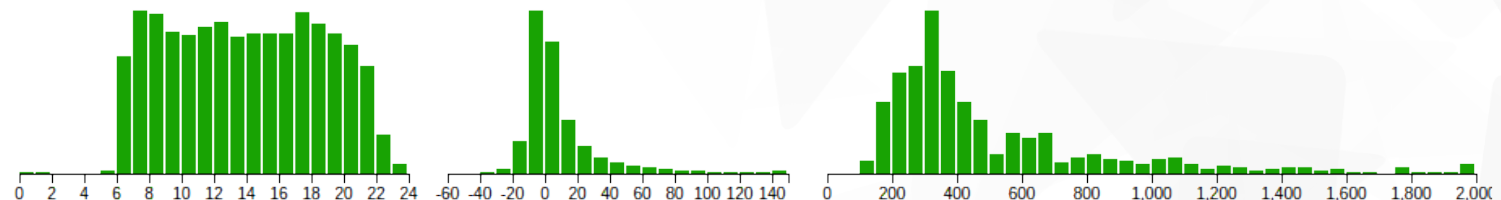
- ▼ Adapté pour l'IoT

```
"year","flows_colorado"  
"1911",18.11  
"1912",21.07  
"1913",15.77  
"1914",24.17  
"1915",14.71
```

```
[  
  {"year": 1911, "flows_colorado": 18.11},  
  {"year": 1912, "flows_colorado": 21.07},  
  {"year": 1913, "flows_colorado": 15.77},  
  {"year": 1914, "flows_colorado": 24.17},  
  {"year": 1915, "flows_colorado": 14.71}  
]
```

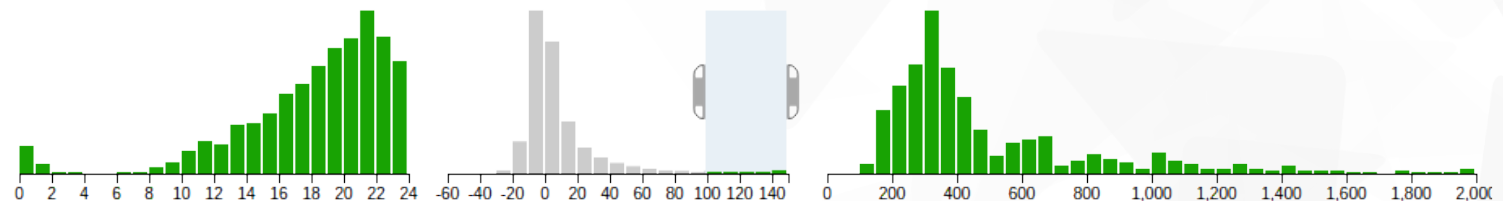
# Étude: exploration des données

- ▼ Observation des données acquises, afin de tirer des conclusions
- ▼ Utilisation des graphiques
- ▼ Utilisation des caractéristiques statistiques simples
- ▼ “Relation entre les données”: l’étude d'influence d'une evolution d'un valeur numérique sur l'évolution d'un autre valeur numérique



# Étude: DC.js

- ▼ Une bibliothèque **JavaScript** pour génération des **graphiques dynamiques** à partir des données
- ▼ Basée sur **D3.js** et **Crossfilter.js**
- ▼ Permet d'utiliser des **filtres multidirectionnelles** afin d'explorer les données, et les relations entre les données
- ▼ Utilise les images vectorielles **SVG**



# Étude: Objectif concret

« Concevoir un *outil graphique* d'exploration des *séries temporelles* provenant de différentes sources d'objets connectés »

# Travail réalisé

- ▼ Deux modules
- ▼ **tsproc** – traitement des séries temporelles
- ▼ **dataexp** – outil graphique

# Travail réalisé: deux modules

- ▼ L'outil est composée des **deux parties** indépendantes:
  - ▼ Un module de **traitement** des séries temporelles
  - ▼ Un module de **visualisation**
- ▼ **Motivation principale**: un module qui traite les données d'un format fixe, et un surcouche qui permet de transformer autres formats en celle de module
- ▼ **Autres motivations**: modularité, ré-utilisabilité, évolutivité



# Travail réalisé: tsproc

- ▼ Time-Series Processor (tsproc) est une **boîte à outil** de traitement des séries temporelles
- ▼ Écrit en: **JavaScript** (Node.js)
- ▼ Format des séries temporelles acceptées: **JSON**
- ▼ Les traitement des séries temporelles possibles:

*sous-échantillonnage*

*fusion des plusieurs vers une*

*interpolation des données manquantes*

*quantification*

*détection de corrélation*

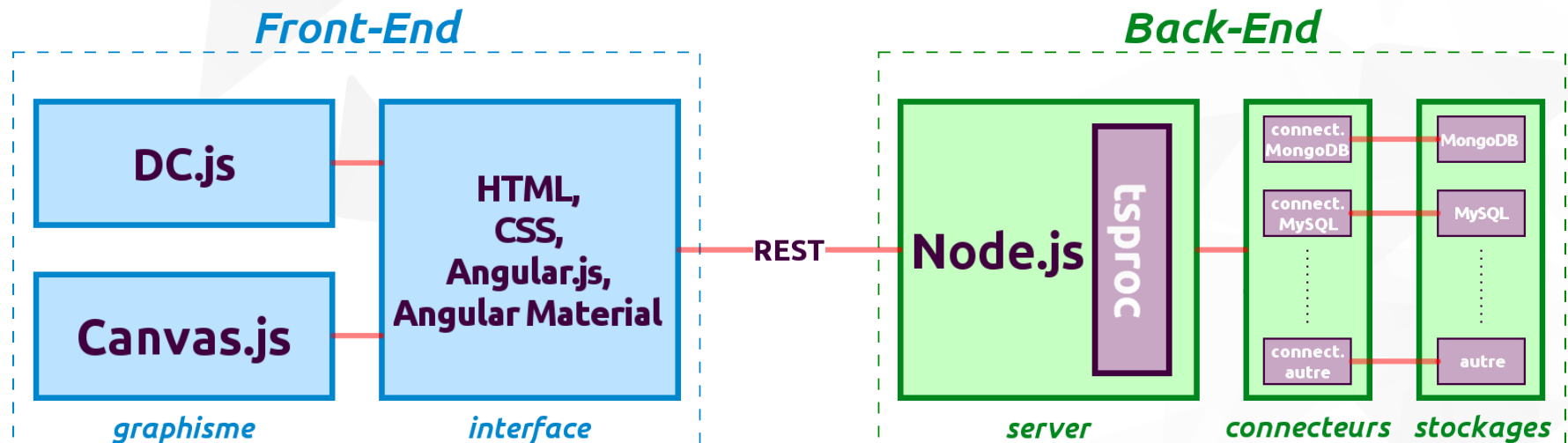
*etc*

# Travail réalisé: dataexp

- ▼ **Data Exploration** (dataexp) est un **outil graphique** qui remplit les tâches suivantes:
  - 1) permet à l'utilisateur d'ajouter ou supprimer les **sources** et choisir les séries temporelles
  - 2) demande de **choisir les mesures à explorer** et spécifier le timestamp
  - 3) **converti** chacun dans un format supporté par **tsproc**
  - 4) offre des **options de traitement** des séries temporelles et offre des **graphiques** pour les explorer

# Travail réalisé: dataexp (2)

- ▼ Types de stockages: MySQL, MongoDB, JSON, CSV
- ▼ Technologies JavaScript utilisées: Node.js, Angular.js, DC.js, Canvas.js
- ▼ Testé pour: Mozilla, Chrome



# Demonstration ...

# Résultats

- ▼ Deux modules écrites en JavaScript
- ▼ Exploration des données via **DC.js**
- ▼ Supporte **plusieurs sources**
- ▼ Lignes de code:

	<u>tsproc</u>	<u>dataexp</u>
<i>Node.js</i>	954	867
<i>Angular.js</i>		1439
<i>HTML</i>		638
<i>Documentation</i>	234	513
Totale:	<b>4645</b>	

## Résultats (2)

### ▼ Adversités:

- ▼ objectif générique
- ▼ gestion de format de timestamp

### ▼ Perspectives:

- ▼ ajouter d'autres sources
- ▼ détection automatique de format de timestamp

# Merci pour votre attention

- ▼ <https://material.angularjs.org/latest/> - module de Angular.js utilisé pour l'interface
- ▼ <https://github.com/dc-js/dc.js/wiki> - graphiques DC.js
- ▼ <http://stackoverflow.com/> - débogage