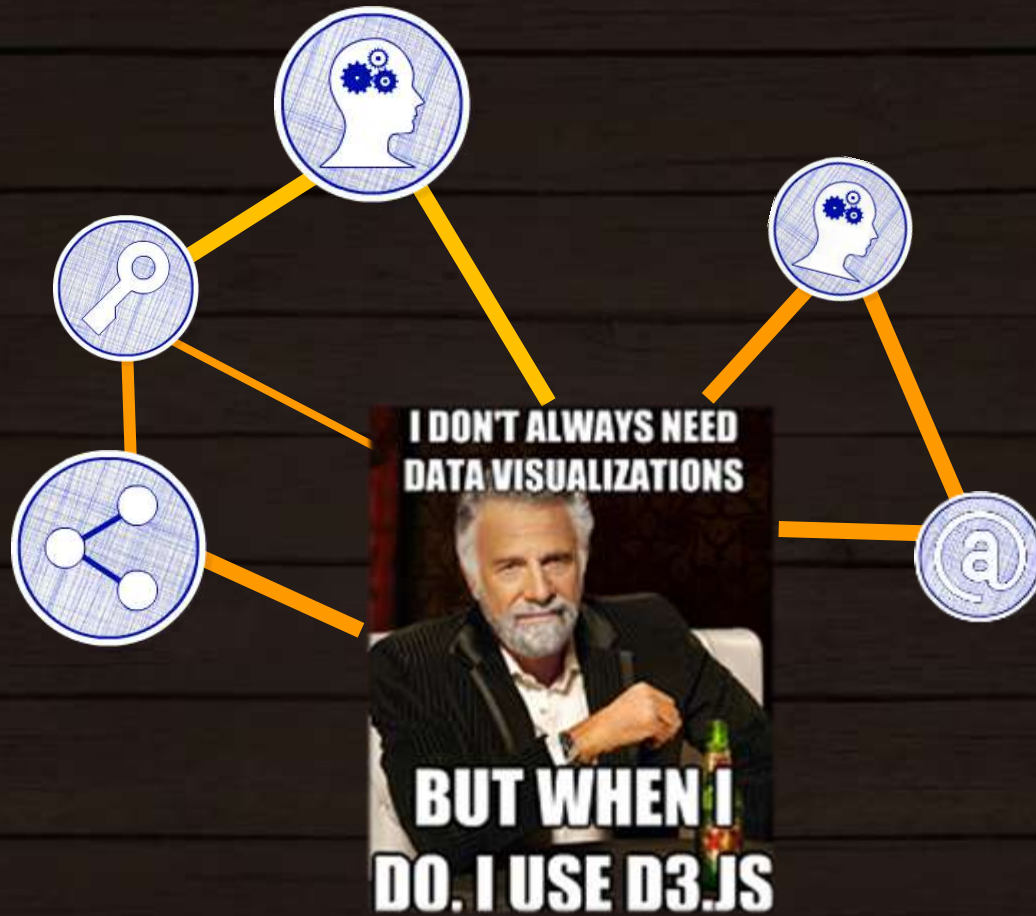


# D3

Une bibliothèque Javascript pour la visualisation de données



Charles Ibrاهيم - @Ibrahimous

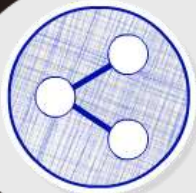
# Agenda de la présentation

## Introduction



### Fonctionnement de D3 par l'exemple

1. Bon, on commence par où ?
2. Les concepts clés dans D3
3. Les concepts moins clés !



### Retour d'expérience sur l'utilisation de D3

1. Faire des choses simples rapidement : check.
2. Pas si simple ?
3. Synthèse : avantages et inconvénients de D3



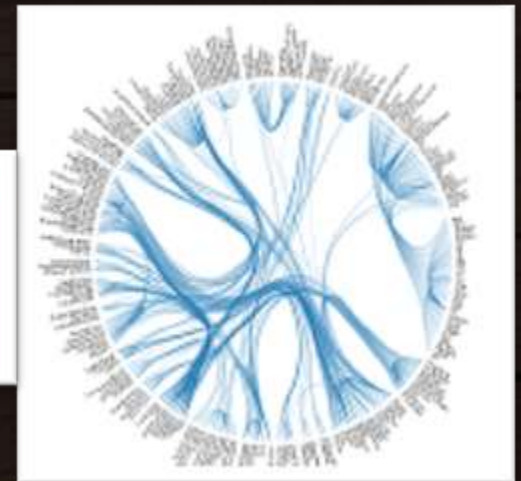
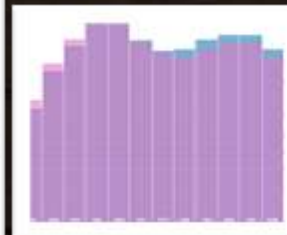
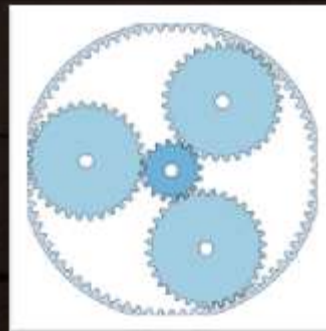
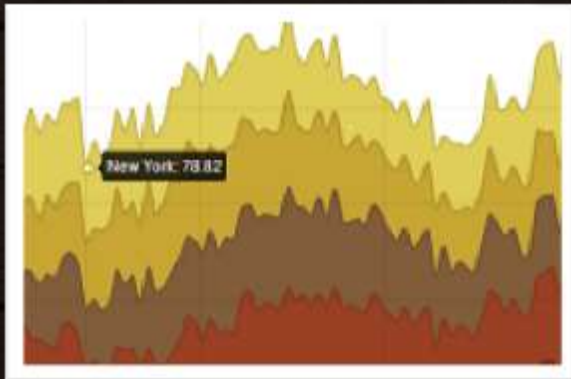
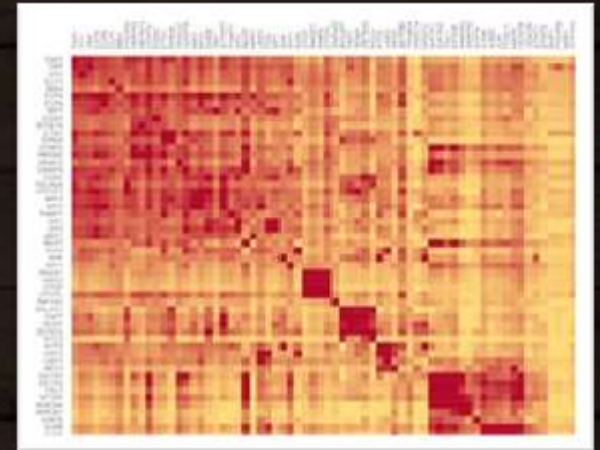
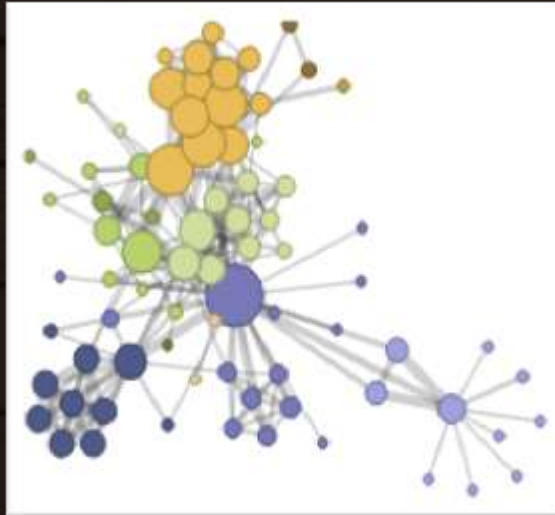
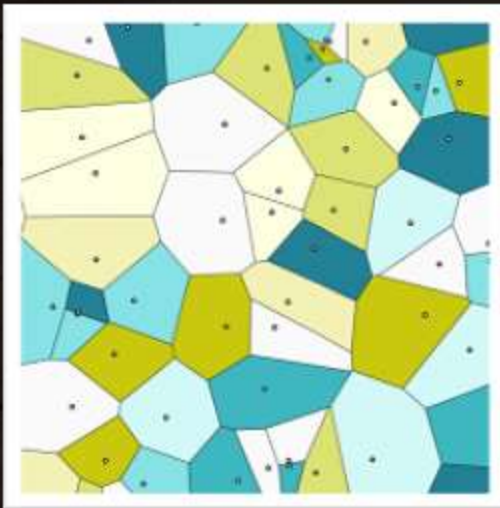
### Focus : la visualisation de données de sécurité

1. ... pour quoi faire ?
2. Exemples et techniques de réalisation
3. Perspectives

## Conclusion

# Introduction : c'est quoi D3 ?

- ✓ Une bibliothèque Javascript
- ✓ ... qui permet de manipuler des éléments HTML et d'y lier des données en masse
- ✓ ... une oeuvre de Mike Bostock : <http://bost.ocks.org/mike/>



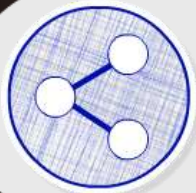
# Agenda de la présentation

## Introduction



### Fonctionnement de D3 par l'exemple

1. Bon, on commence par où ?
2. Les concepts clés dans D3
3. Les concepts moins clés !



### Retour d'expérience sur l'utilisation de D3

1. Faire des choses simples rapidement : check.
2. Pas si simple ?
3. Synthèse : avantages et inconvénients de D3



### Focus : la visualisation de données de sécurité

1. ... pour quoi faire ?
2. Exemples et techniques de réalisation
3. Perspectives

## Conclusion



# ● Bon, on commence par où ?

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="http://d3js.org/d3.v3.min.js">
    </script>
  </head>
  <body>
    <p>Hello!</p>
  </body>
  <script type="text/javascript" src="monsuperscript.js"><script/>
</html>
```

↓

```
d3.select("body")
  .append("svg")
  .attr("width", 50).attr("height", 50)
  .append("circle")
  .attr("cx", 25).attr("cy", 25).attr("r", 25)
  .style("fill", "purple");
```

↙

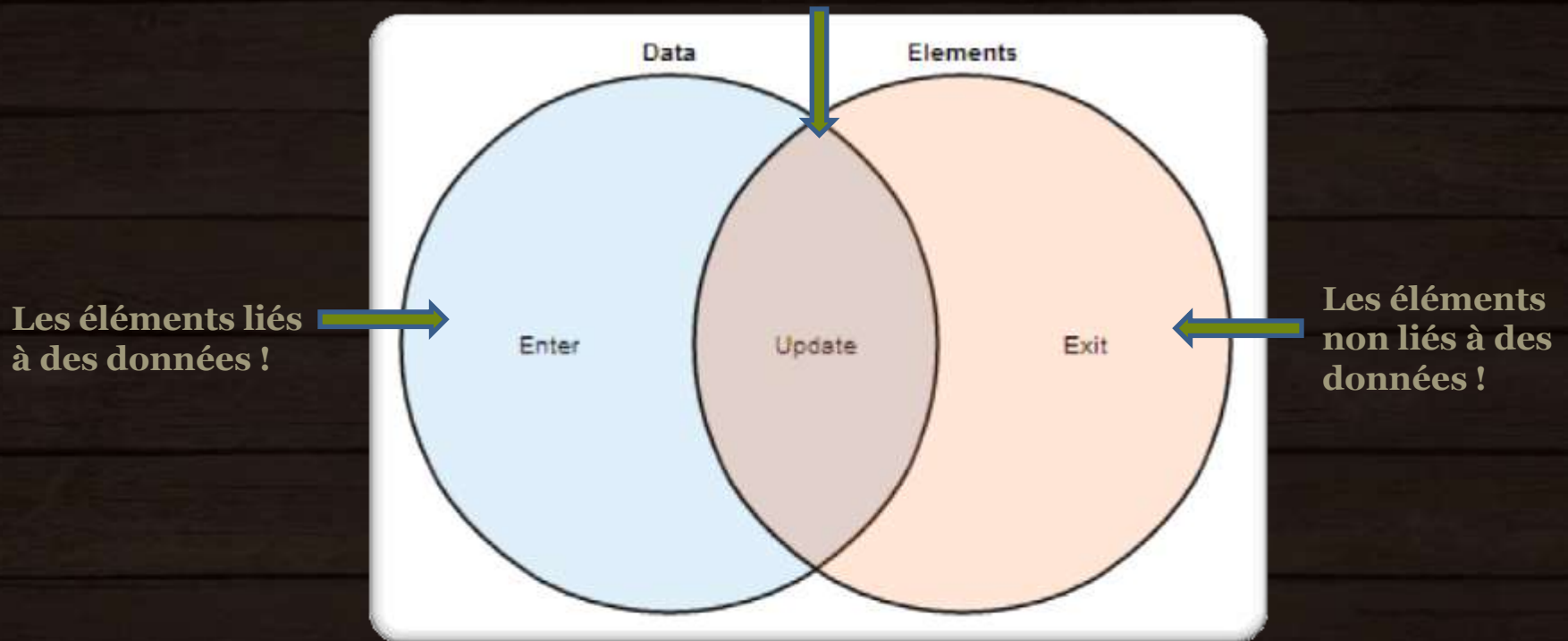
```
<svg width="50" height="50">
  <circle cx="25" cy="25" r="25" fill="purple" />
</svg>
```



# Les concepts clés dans D3

- ✓ Lier des données à des éléments ➡ `“svg.selectAll("circle").data(mesSuperDonnees)”`
- ✓ Afficher les éléments ➡ `.enter().append("circle").attr("cx", function(d) { ... })`
- ✓ Mettre à jour les données

## Les éléments en cours de modification

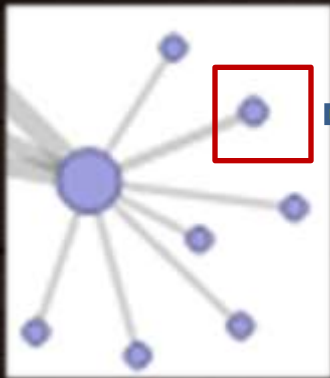


# Les concepts moins clés

✓ Elle est où la fonction update ?

➤ Nulle part !

✓ Cool le diagramme de forces, mais comment je définis la position des cercles ?



```
console.log(d3.selectAll("circle")[0]);  
▼ [circle, circle, circle, circle, circle, circle, circle, circle, circle, circle, circle, circle, circle, circle, circle]  
▼ 0: circle  
  ▼ __data__: Object  
    group: 1  
    index: 0  
    name: "Myriel"  
    px: 446.8070332831753  
    py: 215.8510264101784  
    weight: 10  
    x: 446.6783686543626  
    y: 215.6703471988358
```

➤ T'occupes !

➤ D3 le fait pour toi

✓ Pourquoiuuuuuu quand je charge des données volumineuses elles s'affichent n'importe comment ?

✓ L'asynchronie (WAT ?).

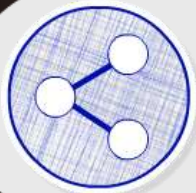
# Agenda de la présentation

## Introduction



### Fonctionnement de D3 par l'exemple

1. Bon, on commence par où ?
2. Les concepts clés dans D3
3. Les concepts moins clés !



### Retour d'expérience sur l'utilisation de D3

1. Faire des choses simples rapidement : check.
2. Pas si simple ?
3. Synthèse : avantages et inconvénients de D3



### Focus : la visualisation de données de sécurité

1. ... pour quoi faire ?
2. Exemples et techniques de réalisation
3. Perspectives

## Conclusion



# ● Faire des choses simples rapidement : *check*.

- ✓ Prise en main rapide : **OK**
- ✓ Simplicité d'utilisation : **OK**
- ✓ Affichage de graphes complexes qui impressionnent votre patron et le client : **OK**
- ✓ Et aussi :
  - ✓ **Documentation** plutôt claire et amusante à lire, nombreux tutoriels
  - ✓ Avantages inhérents à Javascript et aux technologies Web : **portabilité**

# ● Pas si simple ?

```
function redraw() {  
  var rect = chart.selectAll("rect").data(data, function(d) { return d.time; });  
  
  // Enter...  
  rect.enter().insert("svg:rect", "line")  
    .attr("x", function(d, i) { return x(i) - .5; })...;  
  
  // Update...  
  rect.transition()  
    .duration(1000)  
    .attr("x", function(d, i) { return x(i) - .5; });  
  
  // Exit...  
  rect.exit().remove();  
}
```

✓ Fonctions qui font tout toutes seules : euh... ouais.



```
console.log(d3.selectAll("circle")[0]);  
▼ [circle, circle, circle, circle, circle, circle,  
  circle, circle, circle, circle, circle, circle,  
  circle, circle, circle, circle, circle, circle]  
  ▼ 0: circle  
    ▼ __data__: Object  
      group: 1  
      index: 0  
      name: "Myriel"  
      px: 446.8070332831753  
      py: 215.8510264101784  
      weight: 10  
      x: 446.6783686543626  
      y: 215.6703471988358
```

Rappel sur la gestion des objets en Javascript :

<http://anandology.com/presentations/jsfoo-bangalore/inside-javascript-objects>

# ● Synthèses : avantages et inconvénients de D3

- ✓ **Prise en main rapide**
- ✓ **Simplicité d'utilisation pour des POC ou des utilisations dans la ligne des exemples classiques**
- ✓ **C'est joli !**
- ✓ **Documentation claire et assez complète**
- ✓ **Communauté active, nombreux plugins**
- ✓ **C'est du Javascript (portabilité, simplicité)**
- ✓ **Temps réel et plus généralement, personnalisation : pas si évident**
- ✓ **Asynchronie (vive queue.js)**
- ✓ **C'est du Javascript (performances, sécurité, debug)**

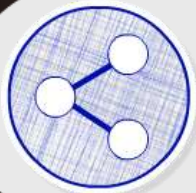
# Agenda de la présentation

## Introduction



### Fonctionnement de D3 par l'exemple

1. Bon, on commence par où ?
2. Les concepts clés dans D3
3. Les concepts moins clés !



### Retour d'expérience sur l'utilisation de D3

1. Faire des choses simples rapidement : check.
2. Pas si simple ?
3. Synthèse : avantages et inconvénients de D3



### Focus : la visualisation de données de sécurité

1. ... pour quoi faire ?
2. Exemples et techniques de réalisation
3. Perspectives

## Conclusion

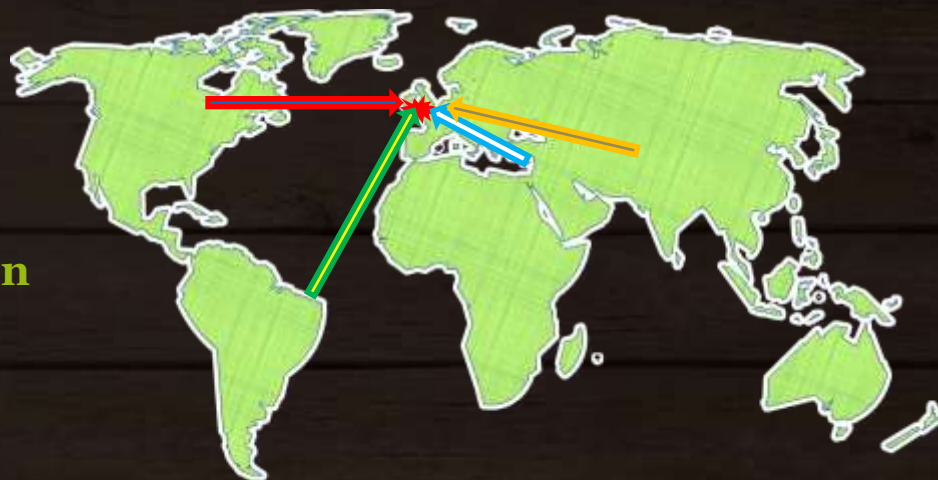
# Visualiser des données de sécurité... pour quoi faire ?



[nvd3.org/examples/stackedArea.html](http://nvd3.org/examples/stackedArea.html)

## ✓ Évident : les stats

- ✓ Nombre d'attaques par pays
- ✓ Type d'attaque
- ✓ ...



## ✓ Un peu moins évident : la réaction en temps (quasi) réel

- ✓ Chef, on détecte des trucs !

## ✓ Plus complexe : l'analyse !

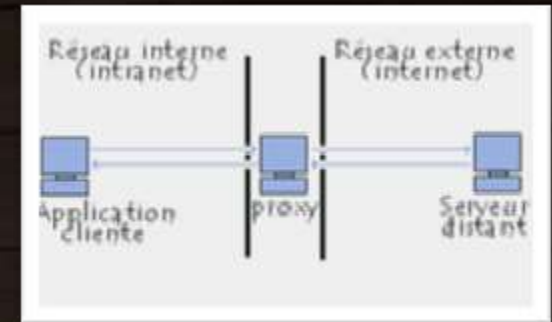
- ✓ Problématique : remettre en contexte une attaque
  - ✓ 3-way handshake qui échoue, requêtes TCP SYN multiples ; DDoS ou téléchargement de photos mal fichus ?
  - ✓ Idem pour des requêtes HTTP dont l'URI est mal formée (trop longue, contenant des caractères étranges...)



# Techniques de réalisation



- ✓ Dans le navigateur : Javascript (D3, jQuery, queue, novus, rickshaw...), HTML, CSS
- ✓ Données récupérées par le navigateur :
  - ✓ des fichiers écrits périodiquement (CSV, JSON, ...)
  - ✓ Un serveur proxy
  - ✓ Une BDD (éventuellement de graphe)
- ✓ Une infrastructure de récupération des données lues par le navigateur (Big Data, ELK, ...)



# ► Perspectives

**Visualiser des données structurées - deux problématiques importantes :**

- ✓ **Visualisation en dimension 2** (+ ou – en dimension 3, peut-être, un jour).
  - ✓ Nécessite de définir des visualisations que l'œil humain puisse percevoir, et que le cerveau humain puisse traiter !
  - ✓ Or, agrégation d'indicateurs => vecteurs à plusieurs dimensions
- ✓ Puis : il n'y a **pas que la puissance de calcul qui compte**
- Représentation des données : problématique plutôt **technique** -> problèmes **mathématiques** (géométrie algébrique, topologie, analyse différentielle)

**Pub : L'analyse de logs dans le contexte de l'Internet des objets > De la visualisation des données : <http://www.observatoire-fic.com/lanalyse-de-logs-dans-le-contexte-de-linternet-des-objets>**

# Conclusion

- ✓ **Un moyen rapide de visualiser des données de façon opérationnelle et marketing**
- ✓ **Interactivité, temps-réel, mise à jour : certaines notions ne sont pas si simples**
  - ✓ **Utiliser D3js : de vraies applications métiers**
- ✓ **Un outil intéressant, mais qui ne doit pas faire omettre une réflexion architecturale globale du traitement des données**

# ● Références

- ✓ Évidemment :
  - ✓ [d3js.org](http://d3js.org) pour une référence des modèles de graphes et une liste de tutoriels
  - ✓ <https://github.com/mbostock/d3/wiki> pour le wiki des fonctionnalités de D3
- ✓ Tutoriel de base, très simple, très concis :  
<https://www.dashingd3js.com/table-of-contents>
  - ✓ Continue en développant des outils au-dessus de D3
- ✓ Comprendre le trio enter, update, exit - Thinking with Joins :  
<http://bost.ocks.org/mike/join/>
- ✓ Tutoriel pour un graphe dynamique avec des cercles qui dansent - Three Little Circles : <http://bost.ocks.org/mike/circles/>
- ✓ Excellent tutoriel sur l'affichage d'un histogramme dynamique :  
<http://strongriley.github.io/d3/tutorial/bar-2.html>
- ✓ D3, la base de nombreuses extensions :  
<http://mikemcdearmon.com/portfolio/techposts/charting-libraries-using-d3>

# *Merci pour votre attention !*

... et classiquement : place aux questions !



&



: @Ibrahimous



&



: Charles Ibrahim