Node.JS

- Environnement pour exécuter du JS en dehors du Navigateur
- *Construit sur le moteur JS V8 de Chrome*
- *Grosse communauté*
- *Full-Stack*

Node.JS

- HTML, CSS, JS, ES6
- *Callbacks, Promesses, Async-Await*

Node.JS

- Intro
- *Installation*
- *Les Fondamentaux de Node.js*
- *Tutoriel Express*
- *Construction d'Apps...*

Browser

- DOM
- Objet Window
- *Apps Intéractives*
- *Pas de Fichiers système*
- *Fragmentation*
- *Modules ES6*

Node.js

- Pas de DOM
- Pas d'objet Window
- Apps coté serveur
- *Fichiers système*
- *Version*
- *CommonJS*

NPM

- NPM INIT - Créé un fichier package.json (manifeste), liste des dépendances
- *NPM INSTALL <package name>*

  Installe un paquet localement (par défaut) et ajoute l'entrée au fichier package.json "dependecies"
- *NPM INSTALL -g <package name>*

  Installe un paquet globalement (accès partout) - SUDO sur Mac/Linux
- *NPM INSTALL -D <package name>*

  Installe un paquet et ajoute l'entrée au fichier package.json DevDepencies (accessible qu'en développement)

Les Modules Intégrés

- OS
- *PATH*
- *FS*
- *HTTP*
- *ect...*

OS

Le module `node:os` fournit des méthodes et des propriétés utilitaires liées au système d'exploitation. On peut y accéder en utilisant :

PATH

Le module `node:path` fournit des utilitaires pour travailler avec des chemins de fichiers et de répertoires.

FS

Le module `node:fs` permet d'interagir avec le système de fichiers.

HTTP

- Le module `node:http` est utilisé pour la création de serveurs.
- *On verra en détail plus tard*

AUTRES SUJETS

- Boucle d'évènements, motifs async, émetteurs et flux d'évènements
- *Concepts principaux*
- Code pré-construit

ÉVÈNEMENTS

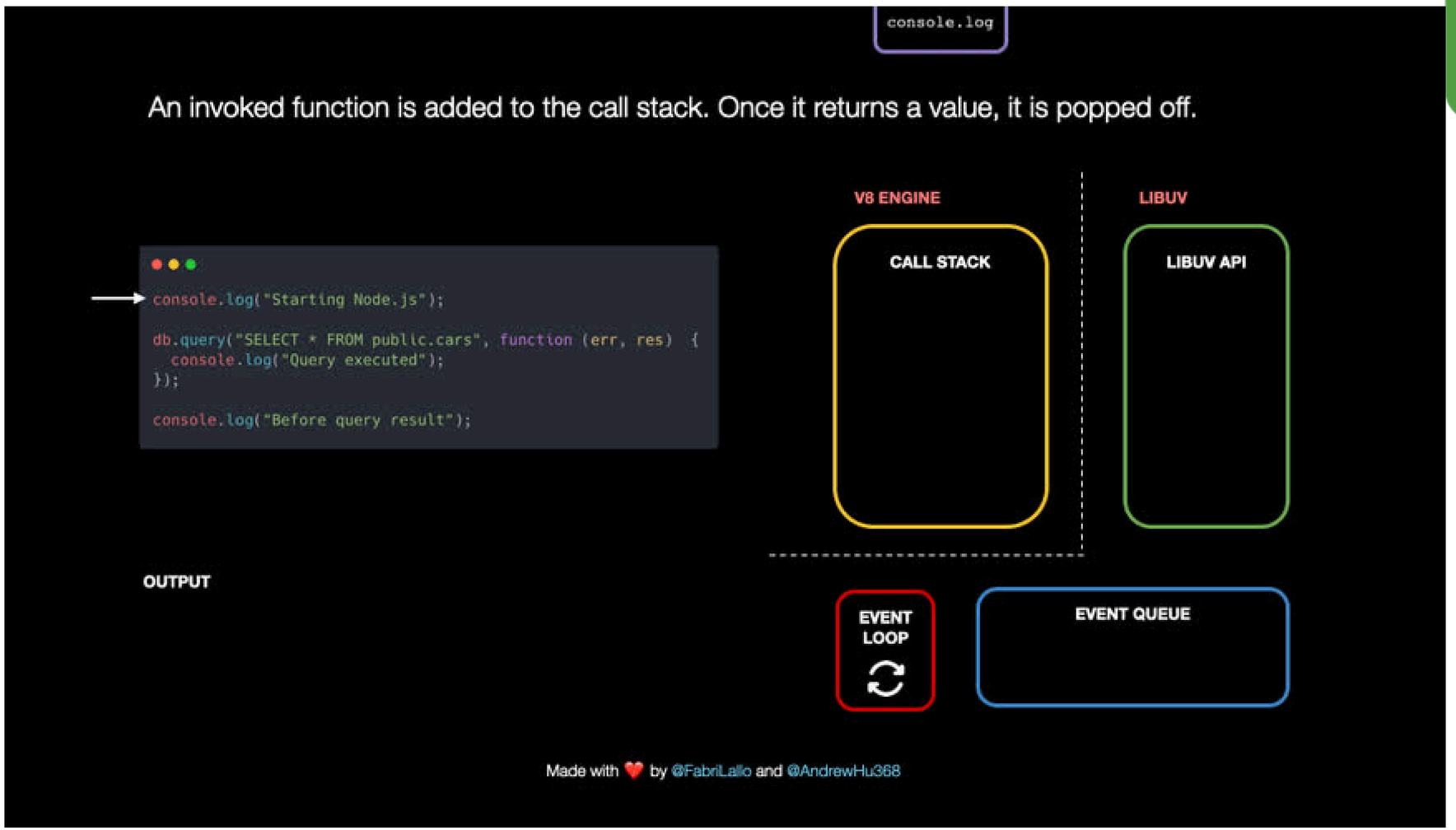
- Programmation évènementielle
- *Largement utilisée dans Node.js*

Qu'est-ce que la boucle d'événements ?

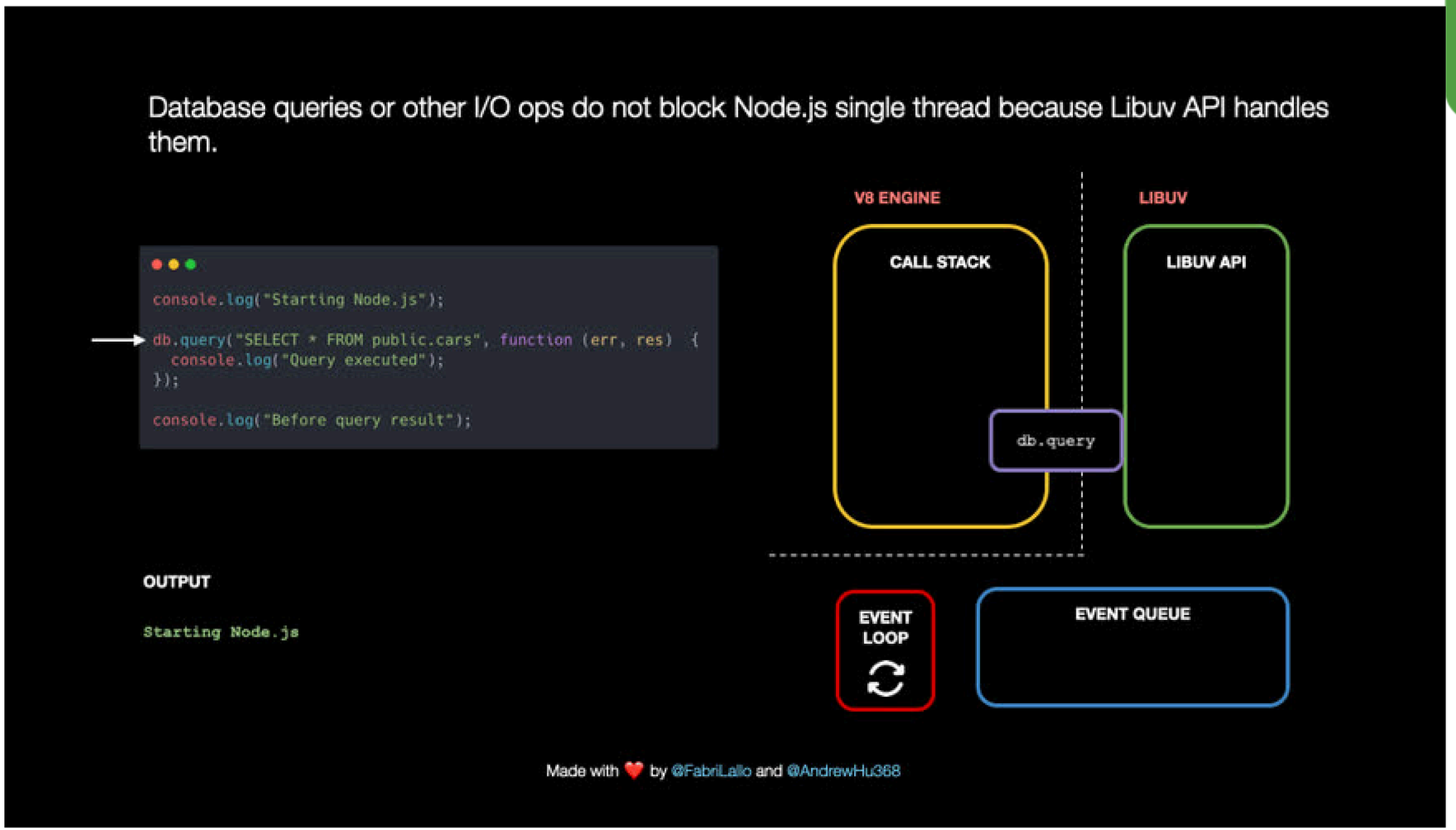
La boucle d'événements permet à Node.js d'effectuer des opérations d'E/S non bloquantes - malgré le fait que JavaScript soit monotâche - en déchargeant les opérations sur le noyau du système chaque fois que cela est possible.

La plupart des noyaux modernes étant multithreadés, ils peuvent gérer plusieurs opérations exécutées en arrière-plan. Lorsque l'une de ces opérations est terminée, le noyau en informe Node.js afin que le callback approprié soit ajouté à la file d'attente pour être éventuellement exécuté. Nous expliquerons cela plus en détail dans la suite de cette rubrique.

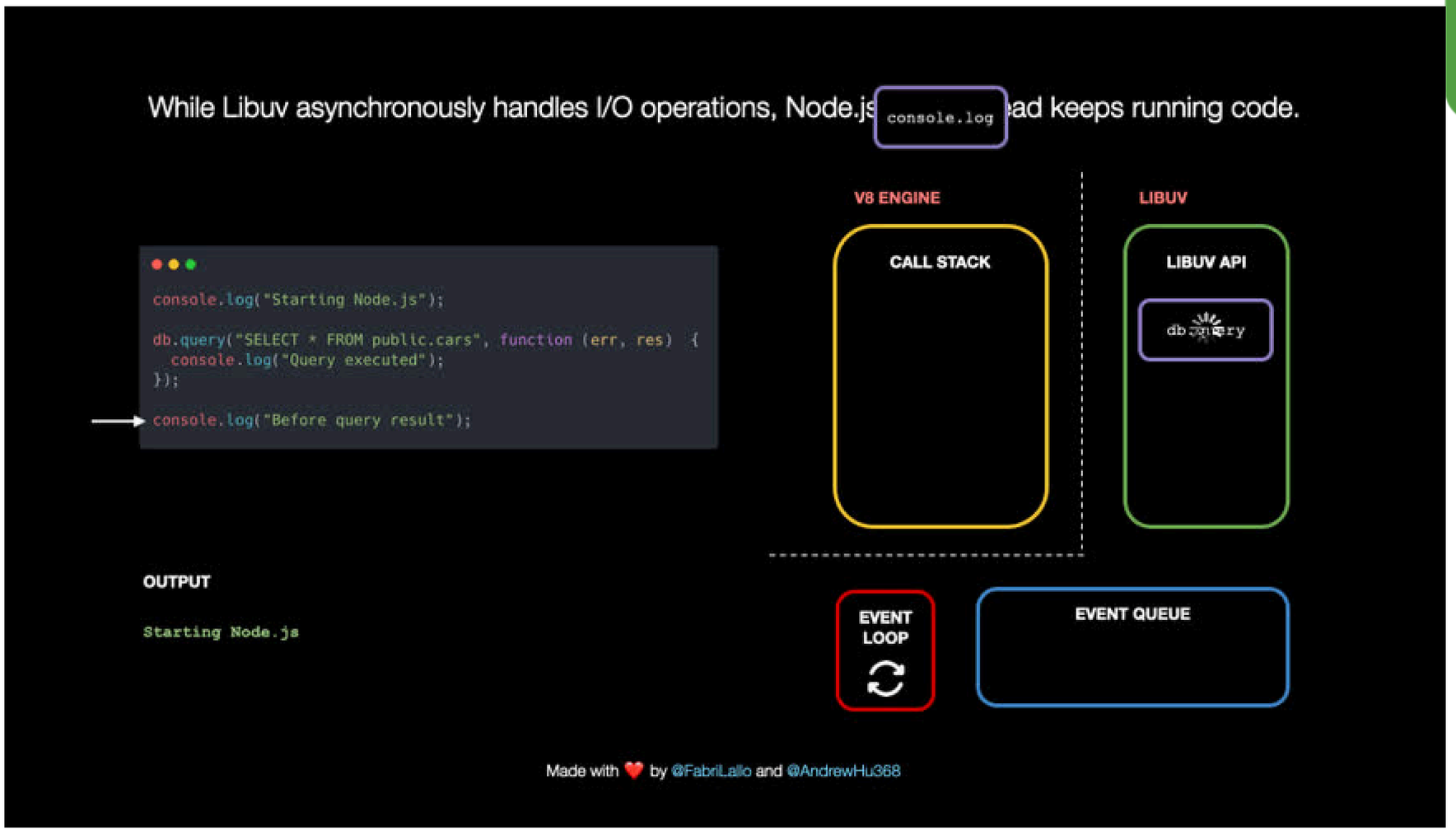
Boucle d'événements



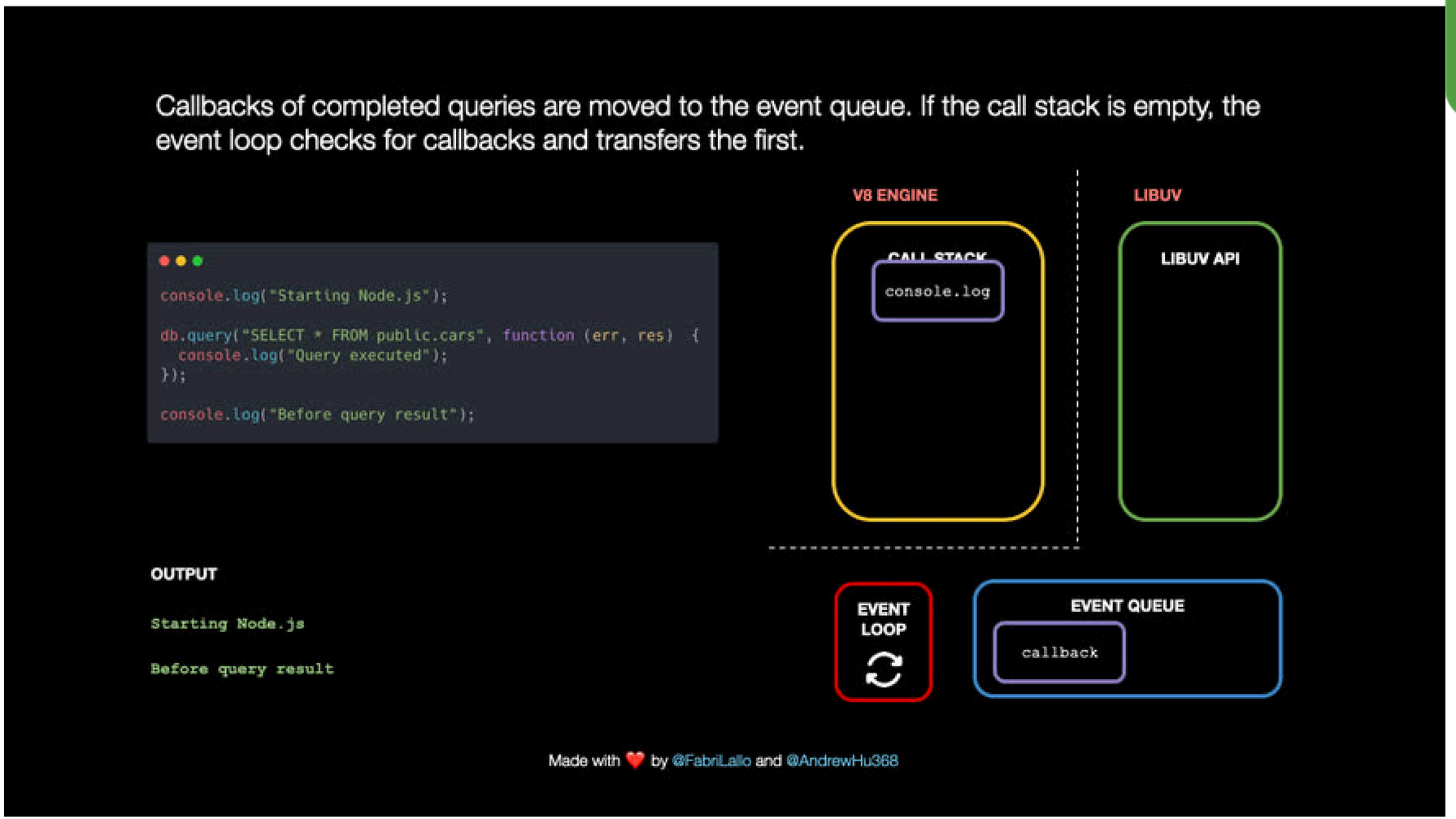
Boucle d'événements



Boucle d'événements



Boucle d'événements



FLUX (STREAM)

Un flux est une interface abstraite permettant de travailler avec des données en continu dans Node.js. Le module `node:stream` fournit une API pour la mise en œuvre de l'interface stream.

De nombreux objets stream sont fournis par Node.js. Par exemple, une requête à un serveur HTTP est une instances de flux.

Les flux peuvent être lisibles, inscriptibles ou les deux. Tous les flux sont des instances d'EventEmitter.

FLUX (STREAM)

- Writeable - Flux dans lequel des données peuvent être écrites.
- *Readable - Flux à partir duquel des données peuvent être lues.*
- *Duplex - Flux qui sont à la fois Readable et Writable.*
- *Transform - Flux duplex qui peuvent modifier ou transformer les données lors de leur écriture et de leur lecture.*

Express.js

API vs SSR

- *API - JSON*
- Envoie des données
- *res.json()*

- *SSR - modèle*
- *Envoie des modèles*
- *res.render()*

MongoDB

- NoSQL, BDD non relationnelle
- *Stocke du JSON*
- *Facile pour démarrer*
- *Hébergement gratuit sur le cloud - Atlas*