



# Saison été : Substratosphere



Découverte de Substrate



# Meetups B612 Crypto Lyon

- Série de meetup initié François Guezengar (Dapps Nation)
- A venir, saison été animé par



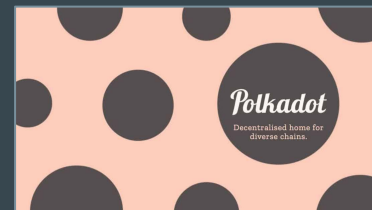
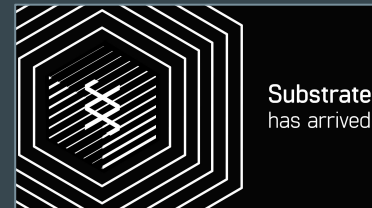
François Branciard



Vladimir Ostapenco

# Saison Été : Substratosphere

- **Découverte de Substrate**
  - B612 - 12/06/2019
- **Découverte de Polkadot**
  - B612 - 19/06/2019
- **Devenez un validateur Polkadot sur le testnet**
  - B612 - 26/06/2019



# Aujourd'hui

- Qu'est-ce Substrate?
- Chaînes spécifiques aux applications
- Qu'est-ce qu'un Runtime?
- Substrate Runtime Module Library (SRML)
- La mise-à-jour forkless
- Template d'un noeud Substrate
- Substrate UI
- Pourquoi Rust?
- Pourquoi Web Assembly?
- Demo: Pile ou face avec Substrate

# Qu'est-ce Substrate?

Substrate est un framework **open source**, **modulaire** et **extensible** pour la création des blockchains.



# Qu'est-ce Substrate?

**Substrate** fournit tous les composants essentiels d'une Blockchain:

- Couche de base de données
- Couche réseau
- Moteur de consensus
- File d'attente de transaction
- Bibliothèque de modules runtime (SRML)



# Qu'est-ce que vous obtenez en plus avec Substrate?

- Connectivité inter-chaîne via Polkadot
- Consensus échangeable à chaud
- Light Client
- Synchronisation de la chaîne
- Pub / Sub WebSocket JSON-RPC
- Bibliothèques JS / Outillage
- Télémétrie
- Mises à jour du runtime sans fork

# Chaînes spécifiques aux applications

## POUR

- **Performance:** machine à états optimisée pour une seule application
- **Sécurité:** la surface d'attaque de la VM est plus petite
- **Souveraineté:** ne dépend pas de la gouvernance de la plateforme
- **Flexibilité:** pas lié aux limitations de la plateforme

## CONTRE

- **Ingénierie:** il est difficile de construire une chaîne à partir de zéro
- **Effets de réseau:** perte d'accès aux données sur d'autres chaînes



# Qu'est-ce qu'un Runtime?

- Le **runtime** est la logique d'exécution de blocs de la blockchain, c'est-à-dire la fonction de transition d'état (State Transition Function).
- Il est composé de modules d'exécution (Runtime Modules).

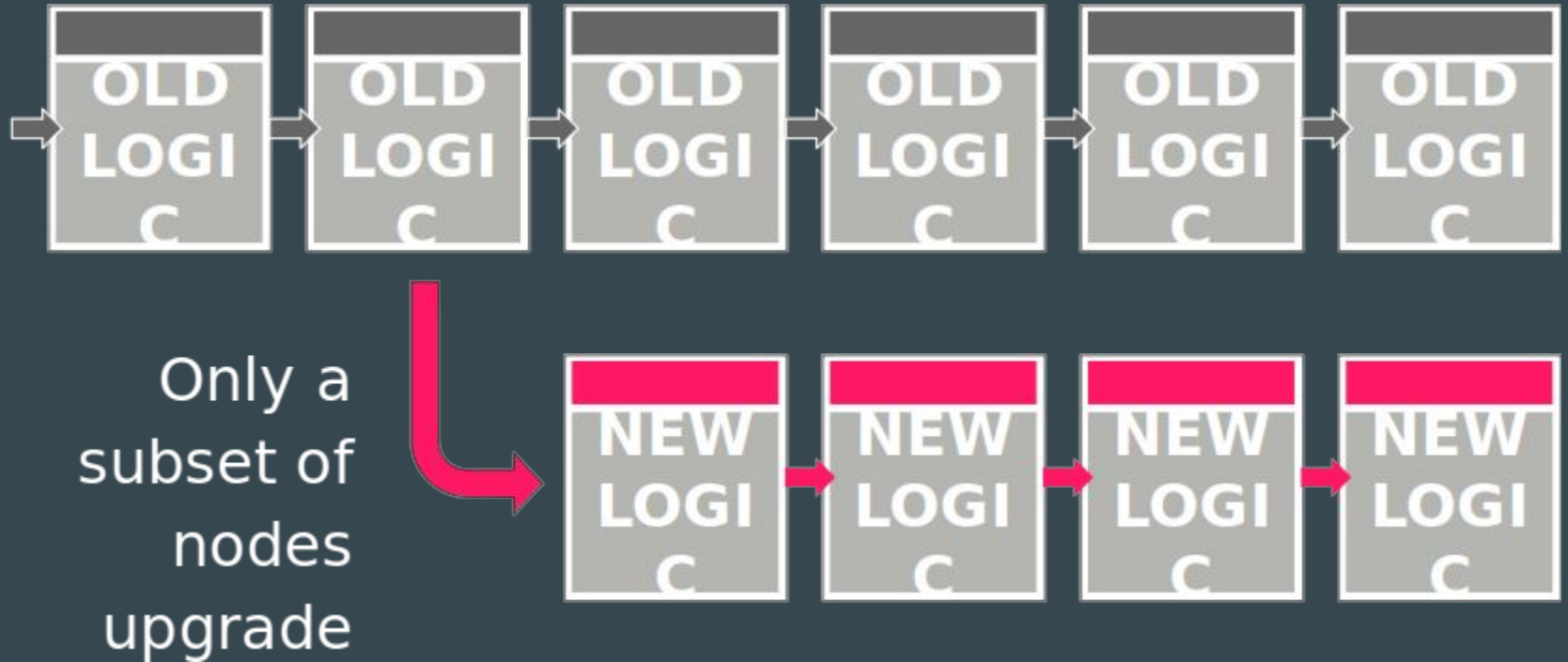


Substrate Runtime Module Library (SRML)			
assets	aura	balances	consensus
contract	council	democracy	executive
fees	grandpa	indices	metadata
session	staking	sudo	system
timestamp	treasury	upgrade-key	and more...

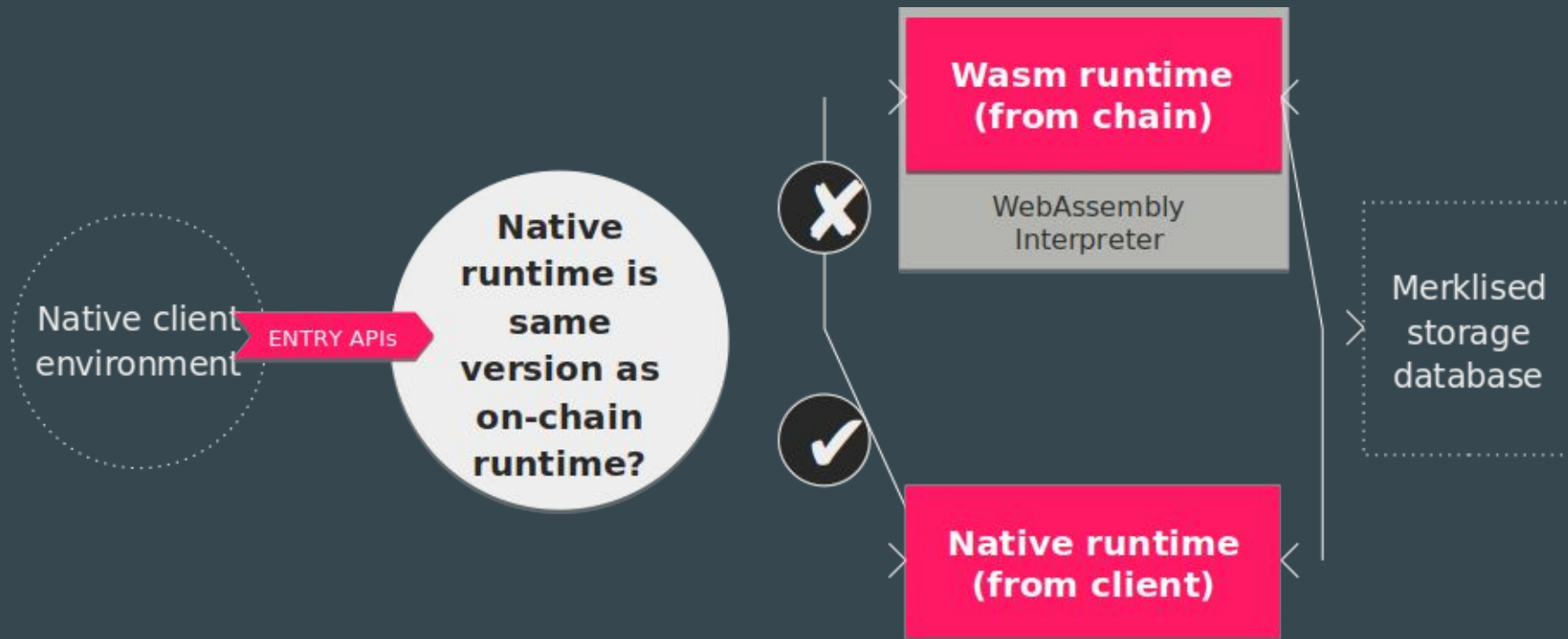
# Substrate Runtime Module Library (SRML)

- SRML est **une collection de modules** runtime
- SRML est livré avec un ensemble de modules prédéfinis qui peuvent être réutilisés dans les runtimes en tant que packages autonomes
- Exemple:
  - Le module "**Balances**" peut être utilisé pour suivre les comptes et les soldes
  - Le module "**Assets**" peut être utilisé pour créer et gérer des actifs fongibles, etc
- Vous pouvez créer votre propre module personnalisé en dérivant
  - du module système
  - d'un ou de plusieurs modules SRML afin d'utiliser leurs fonctionnalités

# La mise-à-jour avec un Fork

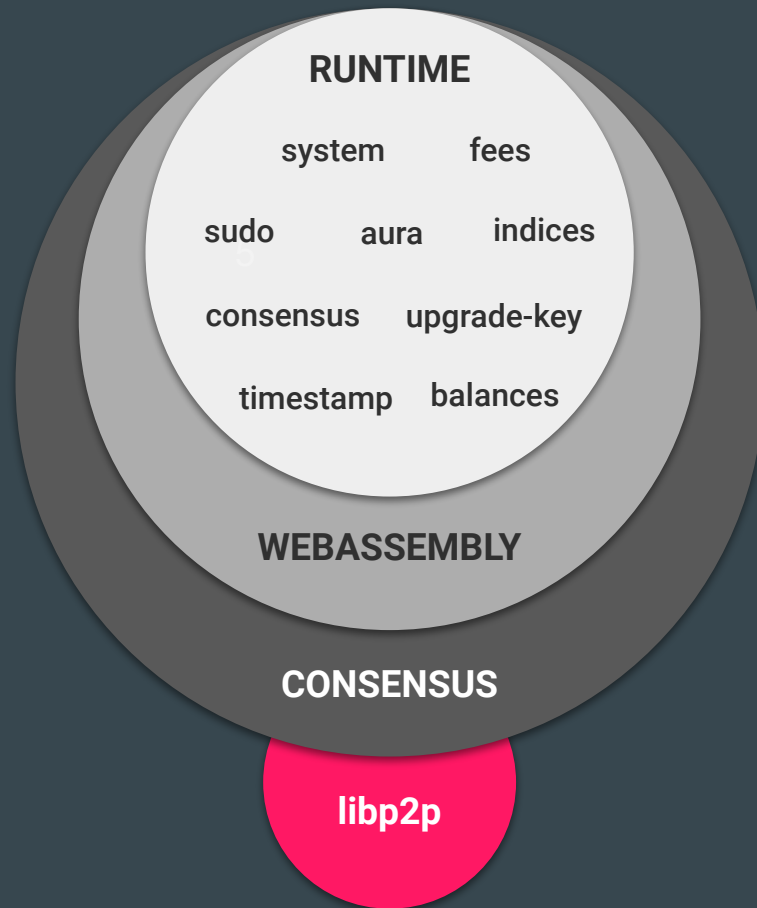


# La mise-à-jour forkless



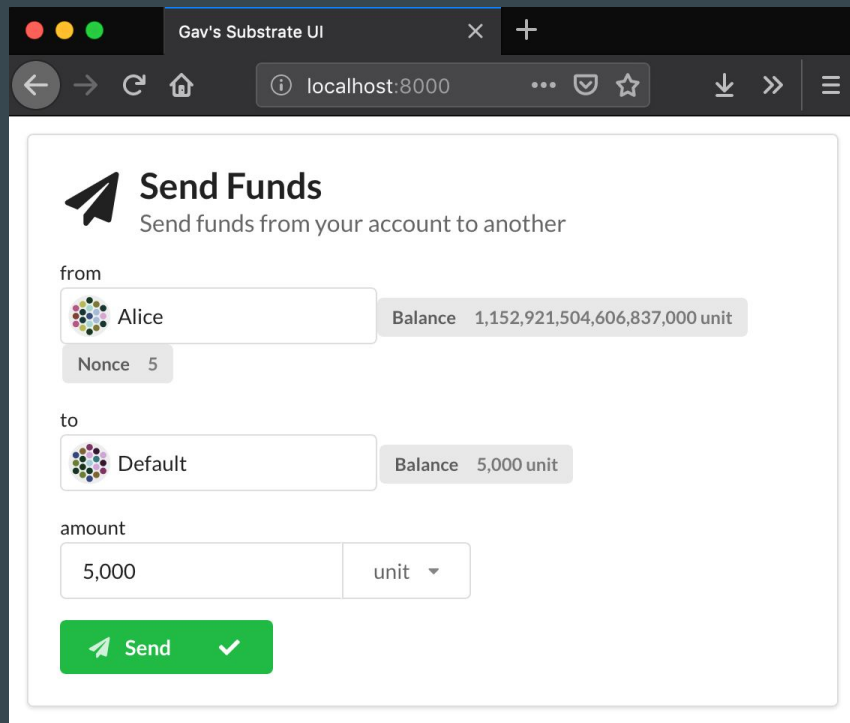
# Template d'un noeud Substrate

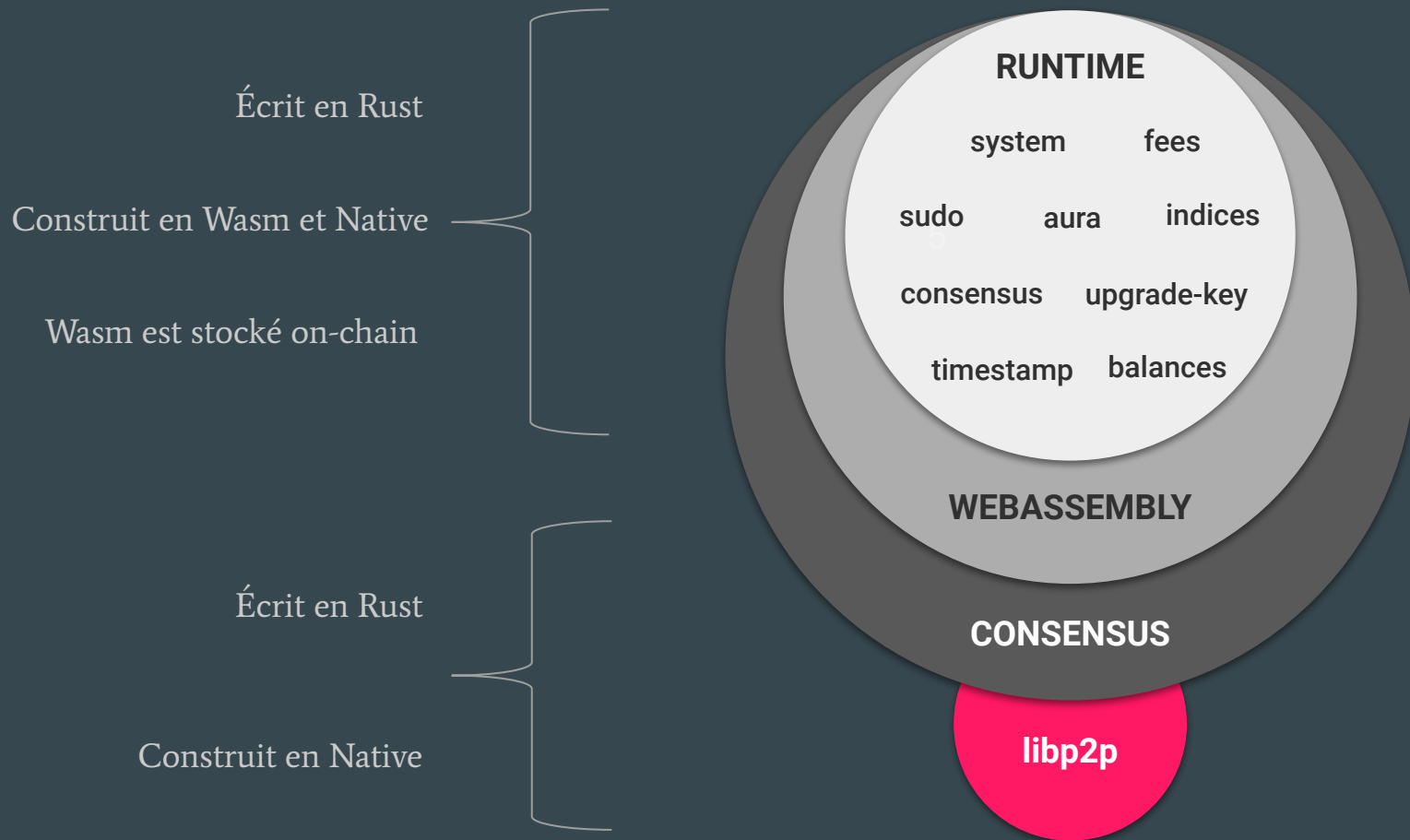
- Un noeud Substrate fonctionnel
- Comprend à partir de SRML:
  - Comptes, soldes, frais, mises à niveau de Runtime, etc.
- Ajoutez et supprimez facilement des modules du SRML
- Créez vos propres modules pour personnaliser les fonctionnalités de votre chaîne



# Substrate UI

- Projet React avec une interface utilisateur pour:
  - Création de comptes
  - Transferts
  - Mises à niveau de Runtime
- Utilise une bibliothèque personnalisée "Bonds"
- Vous pouvez rapidement et ajouter de nouvelles fonctionnalités à l'interface utilisateur





Substate est écrit en Rust.



# Pourquoi Rust?

- Rust est sécurisé
- Rust est rapide
- Rust est léger
- Rust est idiomatique
- Rust est amusant



# Pourquoi Web Assembly?

Wasm est indépendant de la plate-forme

- Wasm est **compact**
- Wasm est **sandboxable**
- Wasm est **deterministe**
- Wasm a la **vitesse d'exécution quasi native**
- Wasm est **bien pris en charge**



# Demo: Pile ou face avec Substrate