BLOCKCHAIN – A DEEPER DIVE

NOVEMBER 2017

RYAN R. FOX
MARC G. SMITH

- BLOCKCHAIN EXPERT CONSULTING
- TECHNICAL ARCHITECTURE & DESIGN
- BUSINESS IMPACT ANALYSIS
- PROOF-OF-CONCEPT & PILOT PROJECT MGMT
- EDUCATION & TRAINING

Ryan R. Fox | ryan@vativ.io | Boston

- Blockchain Professional
- Professional Scrum Master

Marc G. Smith | marc@vativ.io | Minneapolis

- Transaction Processing / Business Process Management expert
- Enterprise software leader: IBM, Lombardi, Trilogy

BLOCKCHAIN – A DEEPER DIVE

- Platforms (compare & contrast)
- Forks / Finality / Governance
- Off-chain Info / Work / Assets
- Smart contracts

…

- Onboarding users to the network
- Cryptographic signing
- Consensus approaches / issues
- Scalability / performance
- Adding nodes to the network

BLOCKCHAIN PLATFORMS

BLOCKCHAIN EVOLUTION

"Distributed Ledger Technology (DLT)"

- **Blockchain 1.0**
  - Bitcoin – system for digital cash exchange
- **Blockchain 2.0**
  - Platforms for "Distributed Apps" – Ethereum, Quorum, …
  - Smart Contracts – programmable business logic for transactions
  - Configurable consensus algorithms
- **Blockchain Enterprise**
  - Enterprise application platforms – Hyperledger & Sawtooth, Microsoft Coco, EOS.IO, Corda …
  - Scaling, Performance, Interoperability, Integrations
  - Specialized hardware and cloud-computing environments

PREVALENT BLOCKCHAIN / DLT PLATFORMS

"Distributed Ledger Technology (DLT)"

	Description	Underlying Technology	Orientation	Consensus Protocol	Smart Contracts
Bitcoin	Purpose-built blockchain system	Bitcoin	Public / crypto-$$	Proof of Work	n/a
Peercoin	Purpose-built blockchain system	Peercoin	Public / crypto-$$	Proof of Stake	n/a
Ethereum	Programmable blockchain platform as-a-service	Ethereum	Public blockchain apps	Proof of Work	Solidity
Quorum	Customizable blockchain platform as-a-service **(JPMorgan / EEA)**	Ethereum	Private enterprise (financial) apps	Pluggable: Byzantine Fault Tolerance (BFT) …	Solidity
IBM Blockchain	Customizable blockchain platform as-a-service on **Bluemix cloud**	Hyperledger	Private enterprise blockchains	Pluggable: Byzantine Fault Tolerance (BFT) …	Go / Java
Oracle Blockchain	Customizable blockchain platform as-a-service on **Oracle Cloud**	Hyperledger	Private enterprise blockchains	Pluggable: Byzantine Fault Tolerance (BFT) …	Go / Java
Intel Sawtooth Lake	Customizable blockchain platform as-a-service on **Intel hardware**	Hyperledger	Private enterprise blockchains	Proof of Elapsed Time (PoET)	Go / Java
R3 Corda	Flexible **DLT** platform – actual blockchains are **not required**!	Corda	Financial DLT applications	Pluggable: Byzantine Fault Tolerance (BFT) …	Java + legal prose
Microsoft Azure	"Open blockchain ecosystem" to host templates for Ethereum, Hyperledger, R3 Corda, BlockApps, … on **Azure**				

FORKS, FINALITY & GOVERNANCE

Forks – Distributed copies of the Blockchain that are not identical replicas

Unintentional

Examples:

- **Race condition** – different transaction blocks are appended simultaneously & race to propagate from opposite sides of the network
- **Network segregation** – temporary network failures prevent appended blocks from propagating across the network

Resolution:

- **Automatic** – One of the forks will grow faster and become favored. Eventually, pending blocks on the slower fork(s) will become orphaned. (**"Finality"** of Blockchain transactions is guaranteed once the orphaned blocks disappear.)

Intentional

Examples:

- **"Bugs"** – accidental insertion of erroneous data or code into the Blockchain
- **Hacking** – malicious insertion of erroneous data or code into the Blockchain

Resolution:

- **Governance** – Intentional forking of the Blockchain into two separate & distinct replicas
  - Bitcoin / "Bitcoin Cash"
  - Ethereum / Ethereum "Classic"

OFF-CHAIN INFO, WORK & ASSETS

Off-chain Info -- **“Oracles”** deliver trusted data

Off-chain Work -- **“Enclaves”** deliver trusted execution

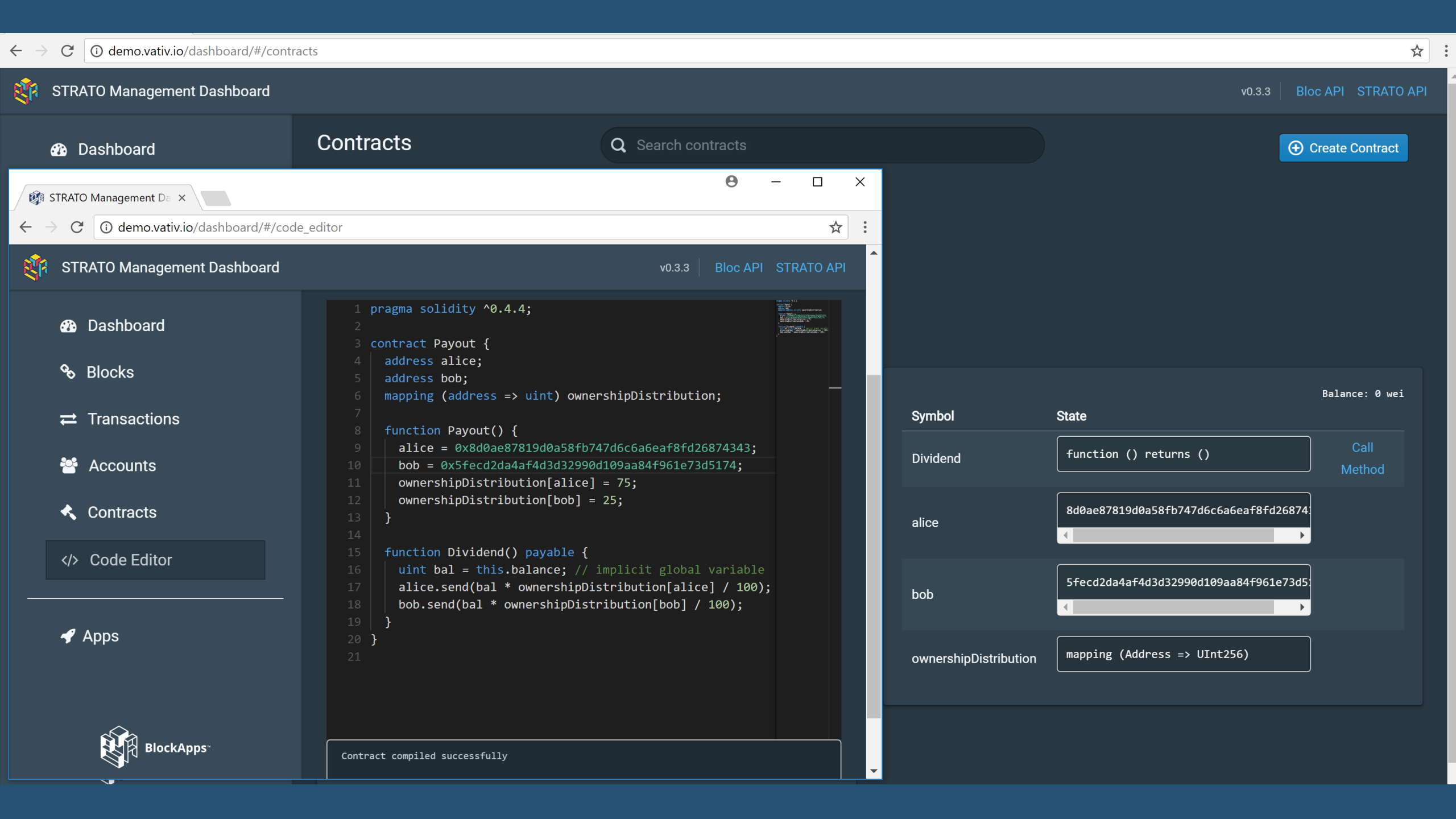
Off-chain Assets -- **Unique identifiers** are trusted “pointers”

SMART CONTRACTS

SMART CONTRACTS

- **Variables:** Contract State
- **Functions:** Contract Actions, Events, and Conditions

… let's look at an example in a simple demo …


demo.vativ.io/dashboard/#/contracts
STRATO Management Dashboard
v0.3.3
Bloc API
STRATO API
Dashboard
Contracts
Search contracts
Create Contract
STRATO Management Da
demo.vativ.io/dashboard/#/code_editor
STRATO Management Dashboard
v0.3.3
Bloc API
STRATO API
Dashboard
Blocks
Transactions
Accounts
Contracts
Code Editor
Apps
BlockApps
pragma solidity ^0.4.4;
contract Payout {
address alice;
address bob;
mapping (address => uint) ownershipDistribution;
function Payout() {
alice = 0x8d0ae87819d0a58fb747d6c6a6eaf8fd26874343;
bob = 0x5fecd2da4af4d3d32990d109aa84f961e73d5174;
ownershipDistribution[alice] = 75;
ownershipDistribution[bob] = 25;
}
function Dividend() payable {
uint bal = this.balance; // implicit global variable
alice.send(bal * ownershipDistribution[alice] / 100);
bob.send(bal * ownershipDistribution[bob] / 100);
}
}
Contract compiled successfully
Balance: 0 wei
Symbol
State
Dividend
function () returns ()
Call Method
alice
8d0ae87819d0a58fb747d6c6a6eaf8fd26874
bob
5fecd2da4af4d3d32990d109aa84f961e73d5
ownershipDistribution
mapping (Address => UInt256)

OTHER QUESTIONS?

THANK YOU VERY MUCH!