Bringing IOTA Distributed Ledger Technology (DLT) into Yocto/OpenEmbedded

Bernardo A. Rodrigues
Philipp Blum (IOTA Foundation)
Presenters

• Bernardo A. Rodrigues
  meta-iota Maintainer
  bernardoaraujor@gmail.com

• Philipp Blum
  Developer Advocate (IOTA Foundation)
  philipp.blum@iota.org
Table of Contents

• What is IOTA?
• IOTA Nodes
• meta-iota
• IOTA Ecosystem Development Fund
What is IOTA?

Context
Distributed Ledger Technologies

Blockchain

Tangle (DAG - Directed Acyclic Graph)
Tangle (DAG)

More Activity = More Validation

Each Vertex represents a transaction (squares)

Each Edge represents an approval (lines)
Zero Fee Transactions

No mining = No fees = Zero fee micro-transactions
IOTA Foundation

- Non-Profit Foundation registered in Berlin
- ~100 employees in 17 countries
- Funded through donations from IOTA Token holders, Research Grants and Project-based corporate financial support
IOTA Foundation

Vision

Objective
Grow a vibrant & engaged open source community
Build a decentralized and self-regulated P2P network
Adoption & real world implementation

Purpose

Research
Produce a strong theoretical foundation for the core tech, and enable ambitious future use cases in the economy of things ecosystem.

Technology
Develop production-ready software for the community, partners and ecosystem to use and expand upon.

Education
Educate and promote technologies and use cases for new generations to understand.

Standardization
Standardize and ensure the maturity and widespread adoption of the economy of things.

Principles

Co-Creation, Transparency, Decentralization
IOTA Foundation: Collaborations & Partnerships

source: http://iotaarchive.com
IOTA Nodes

Pre Coordicide vs Post Coordicide
IOTA Nodes

• DLT Node:
  • transaction relay
  • ledger copy
IOTA Nodes

- Ethereum, Bitcoin, etc: Nodes on the Cloud (↑ hw resources)
- IOTA: Nodes on the Edge (↓ hw resources)

Towards Open Collaboration:
The Linux Foundation & IOTA Foundation join forces through LF Edge

https://blog.iota.org/towards-open-collaboration-1926e94514b8
Coordicide

- To make it possible for the network to grow and protect it against certain attacks, IOTA currently relies on a coordinator.

- The coordinator checkpoints valid transactions, which are then validated by the entire network.

- The coordinator is being run by the IOTA Foundation.

- Removing the Coordinator from the IOTA network will realize a long sought after goal in the field of DLT: scalability without centralization.

- Coordicide: the death of the Coordinator.
Pre Coordicide vs Post Coordicide

- Pre-Coordicide Node implementation:
  - IRI (Java)
  - cIRI (C)

- Coordicide Proof of Concept Node implementation:
  - GoShimmer (Go)

- Post-Coordicide Node implementation:
  - Bee (Rust)
  - Hornet (Go)

Since Coordicide is still a topic under R&D, meta-iota focuses on Pre Coordicide (for the moment).
cIRI

- low level implementation of an IOTA node in C
- Users to become part of the IOTA network:
  - transaction relay
  - network information provider
- JSON-REST HTTP interface
- Suited for Embedded (SoC, SoM):
  - RAM: down to ~140MB RAM for solid node, ~500MB while syncing
cIRI: Bazel

- IF development team chose Bazel as build system for cIRI
- I borrowed the Bazel recipe and bbclass from meta-tensorflow
- Plans to switch to CMake
ciri_0.1.0.bb

- https://github.com/bernardoaraujor/meta-iota/blob/master/recipes-iota/ciri/ciri_0.1.0.bb
Let's ping the cIRI node on the BBB

```bash
$ curl http://104.155.135.221:14265/ \
  -X POST \ 
  -H 'Content-Type: application/json' \ 
  -H 'X-IOTA-API-Version: 1' \ 
  -d '{"command": "getNodeInfo"}'
```
CClient

- IOTA client library implementation in C.
- Recipe exports libcclient.a into the target rootfs/sysroot.
- CMake support
- Patch CMakeLists.txt to avoid the ExternalProject_add feature of CMake
- Recipe for c-iota-workshop repository as an example of how to integrate with libcclient
libcclient_1.0.0.bb

- https://github.com/bernardoaraujor/meta-iota/blob/master/recipes-iota/cclient/libcclient_1.0.0.bb
Playing around with c-iota-workshop

• Install Bazel:
  https://docs.bazel.build/versions/master/install.html

• Clone repo:
  $ git clone https://github.com/iota-community/c-iota-workshop

• Run an example:
  $ cd c-iota-workshop
  $ bazel run -c opt examples:[EXAMPLE_NAME]

• Following examples are available:
  hello_world
  send_hello
  receive_hello
  generate_address
  check_balances
  send_tokens
iota.go

- IOTA Go API Library allows:
  - Create transactions
  - Sign transactions
  - Interact with an IRI node

- Recipe written, although more testing is needed for validation.

- Recipe lists all golang package dependencies explicitly.

- Recipe for go-iota-workshop repository as an example of how to integrate with iota.go library
iota.go_1.0.0.bb

- https://github.com/bernardoaraujor/meta-iota/blob/go-dev/recipes-iota/iota.go/iota.go_1.0.0.bb
Playing around with go-iota-workshop

- Install Golang (1.10+)
  https://golang.org/doc/install

- Clone repo and download dependencies:
  $ git clone https://github.com/iota-community/go-iota-workshop
  $ cd go-iota-workshop; go mod download

- Run an example:
  $ go run iota_go_[EXAMPLE_NAME]/main.go

- Following examples are available:
  helloworld
  send_data
  receive_data
  create_address
  check_balance
  send_tx
  receive_tx
  zmq
iota.lib.py / PyOTA

• Official Python library for the IOTA Core.
• Implements both the official API, as well as signing, bundles, utilities and conversion.
• Python 3.6, 3.5 and 2.7.
• inherit setuptools
• Integration is planned for the near future
• https://github.com/iotaledger/iota.lib.py
Playing around with python-iota-workshop

- Install Python 3 and PIP
  https://www.python.org/downloads/

- Clone repo and download dependencies:
  $ git clone https://github.com/iota-community/python-iota-workshop
  $ cd python-iota-workshop; pip install -r requirements.txt

- Run an example:
  $ python code/[EXAMPLE_NAME].py

- Following examples are available:
  e01_hello_world.py  e04_generate_address.py  e07_send_data.py
  e02_send_hello.py  e05_check_balance.py  e08_receive_data.py
  e03_receive_hello.py  e06_send_tokens.py  e09_zmq_listen.py
IOTA CLI App

- Command Line wallet and node management tool.
- It is implemented in nodejs, and it’s available as a npm package.
- To be integrated with the help of devtool npm functionality.
- Integration planned for the near future.
- https://github.com/iotaledger/cli-app
- https://wiki.yoctoproject.org/wiki/TipsAndTricks/NPM
recipes-support

- In order to fulfill dependencies, I had to write a few support recipes.
  - `nanopb_0.3.9.3.bb`: small code-size Protocol Buffers implementation in ansi C. Especially suitable for use in microcontrollers, but fits any memory restricted system
  - `keccak_git.bb`: keccak sponge function family including SHA3 implementation. Recipe needs improvement to support more architectures
  - `logger_4.0.0.bb`: simple logging facility for the C language
  - `libzmq_4.3.2.bb`: ZeroMQ core engine in C++
- Extra contribution to the OE community.
Future of meta-iota (2020-21)

• Bee
  • Post Coordicide Reference Implementation
  • Official IOTA Foundation
  • Rust (meta-rust and meta-rust-bin?)

• Hornet
  • Post Coordicide Implementation
  • Community based (EDF)
  • Go
Ecosystem Development Fund

Boards for Proof-of-Concept
IOTA Ecosystem Development Fund

- The IOTA EDF will allow me to validate Proof-of-Concepts on a few different boards with potential for IOTA Industrial applications.

- There is a big interest for FPGA projects in the IOTA Community. This is due to the Quorum Based computations, as well as accelerated Proof-of-Work (PoW), Address Generation and Signing.

<table>
<thead>
<tr>
<th>Board</th>
<th>Manufacturer</th>
<th>Comment</th>
<th>OpenEmbedded BSP Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>STM32MP157C-DK2</td>
<td>STMicroelectronics</td>
<td>The discovery SBC for STMicroelectronics STM32MP1 Series microprocessors</td>
<td>meta-st-stm32mp</td>
</tr>
<tr>
<td>Colibri iMX6 Solo SoM + Viola Carrier</td>
<td>Toradex</td>
<td>Toradex is a swiss manufacturer of Industrial-grade System on Modules.</td>
<td>meta-freescale-3rdparty</td>
</tr>
<tr>
<td>Zynq-7000 SoC ZC702 Evaluation Kit</td>
<td>Xilinx</td>
<td>The Zynq-7000 is a SoC+FPGA with great potential to accelerate PoW, Mini-PoW, Address Generation and Signing, as well as future Qubic implementations.</td>
<td>meta-xilinx</td>
</tr>
</tbody>
</table>
## IOTA Ecosystem Development Fund

<table>
<thead>
<tr>
<th>Board</th>
<th>Manufacturer</th>
<th>Comment</th>
<th>OpenEmbedded BSP Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE10-Nano Development Kit</td>
<td>Terasic Technologies</td>
<td>The E10-Nano Development Kit is built around the Intel/Altera CycloneV SoC+FPGA. Also great potential to accelerate PoW, Mini-PoW, Address Generation and Signing, as well as future Qubic implementations.</td>
<td>meta-de10-nano meta-altera</td>
</tr>
<tr>
<td>BeagleBone Black</td>
<td>Texas Instruments</td>
<td>The most popular SBC in the Yocto Community.</td>
<td>meta-yocto-bsp meta-ti meta-beagleboard meta-bbb</td>
</tr>
<tr>
<td>DragonBoard 410c</td>
<td>96Boards</td>
<td>SBC with a Qualcomm Snapdragon 400</td>
<td>meta-qcom</td>
</tr>
<tr>
<td>Orange Pi Zero</td>
<td>Orange Pi</td>
<td>Popular small SBC with an AllWinner H2 chip.</td>
<td>meta-sunxi meta-allwinner-hx</td>
</tr>
<tr>
<td>Raspberry Pi Zero W</td>
<td>Raspberry Pi Foundation</td>
<td>Miniature version of the RPi, with Wireless support.</td>
<td>meta-raspberrypi</td>
</tr>
</tbody>
</table>
Thank you!