Pushing the Raspberry Pi: Ethereum on Pi and Fluree on Pi. Low-cost and Open Source Enterprise Computing for Fun, Learning, and Training.

Prepared for the 2021 CAPDA ICT Africa SYMPOSIUM Conference in Yaounde, Cameroon, Central Africa UTC+1

William Favre Slater, III
Chicago, Illinois
United States of America
UTC-5 Central Daylight Time

August 24 - 26, 2021

Slater Technologies

PULSE CONNECT

Disclaimer

This presentation will be delivered in August 2021 at the request of the organizers of the TIC Afrique Symposium CAPDA Conference in Yaounde & Libreville, Cameroon.

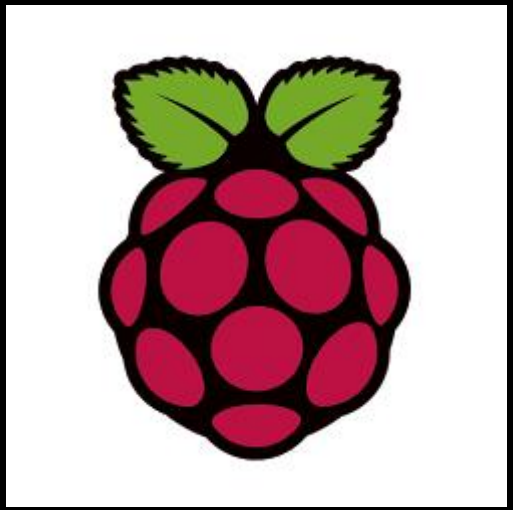
The content presentation is from William Favre Slater, III, a private U. S. Citizen, & President of Slater Technologies, Inc., and his research sources. It was prepared and presented with Good Will as a Service to Benefit the Good Citizens of Republic of Cameroon, and not as a representative of the U.S. Government or any other organization.

Mr. Slater only represents himself in his dealings with the Citizens of Cameroon and is solely responsible for his comments, content, and research.

Finally, this is NOT A SALES PRESENTATION. It is a market assessment of existing capabilities and a discussion about low-cost options and the Art of the Possibilities.

Agenda

- Introduction
- Why this information is important?
- Computing Evolution
- What Is the Internet of Things?
- Personal Computing, 1985 vs. 2021
- What is Raspberry Pi?
- What is Ethereum?
- Putting Ethereum on Raspberry Pi
- What is Fluree?
- Putting Fluree on Raspberry Pi
- Conclusion



Introduction

- **WhoAmI?** William Favre Slater, III
- An American Citizen and a former U.S. Air Force Computer System Staff Officer
- Over 40 years in Information Technology
- Several Specialties: Cybersecurity, Project Management, Program Management, Programming (43 languages), Databases, Cloud, Networking, Services Management, System Design and Implementation, Troubleshooting, Identity Management, Blockchain, Technical Writing, Governance, Risk Management and Compliance, etc.
- Three graduate degrees, including an M.S. in Cybersecurity and an MBA and over 80 Professional IT Certifications
- Over 30 years of teaching experience
- Married since 2000, to Ms. Joanna Roguska
- This is my ***fourth time*** to present to our Dear Cameroon Colleagues, since July 2020.

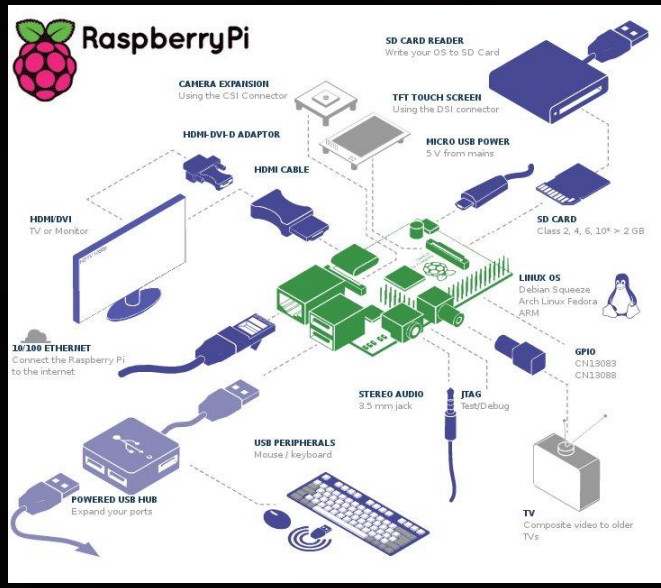
Introduction

- This presentation is about setting up two different powerful FREE Opensource software systems on the Raspberry Pi, which is a low-cost, powerful Internet of Things (IoT) Platform

- Ethereum is one of these two distributed, decentralized systems.

- Fluree is the other two distributed, decentralized systems.

- The experience of setting up and learning either or both of these systems can greatly increase an individual's value in the IT market.

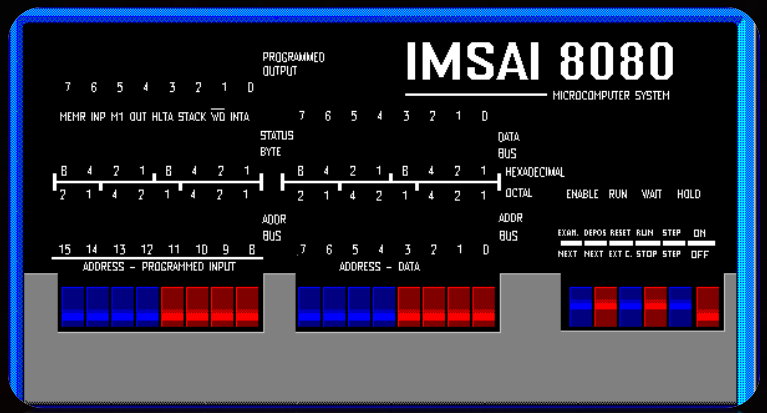


Why Is This Information Important?

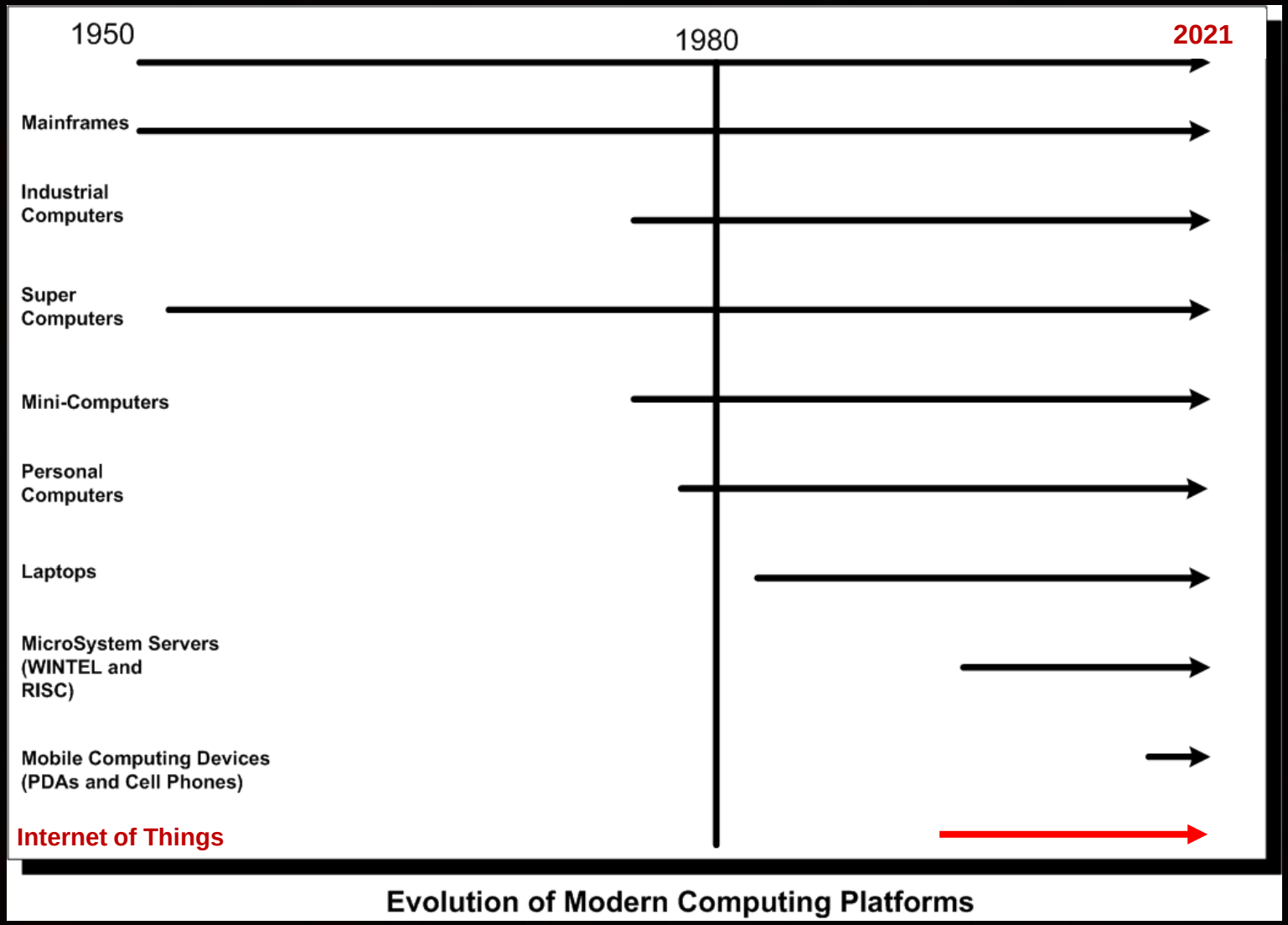
- With the low cost and the incredible power of the Raspberry Pi this may be the best computing platform to learn and get marketable experience.

- Availability of FREE Open Source software like Ethereum and Fluree offer easy and a compelling options to get valuable experience for a career in Information technology.

Computing Evolution



Computing Evolution



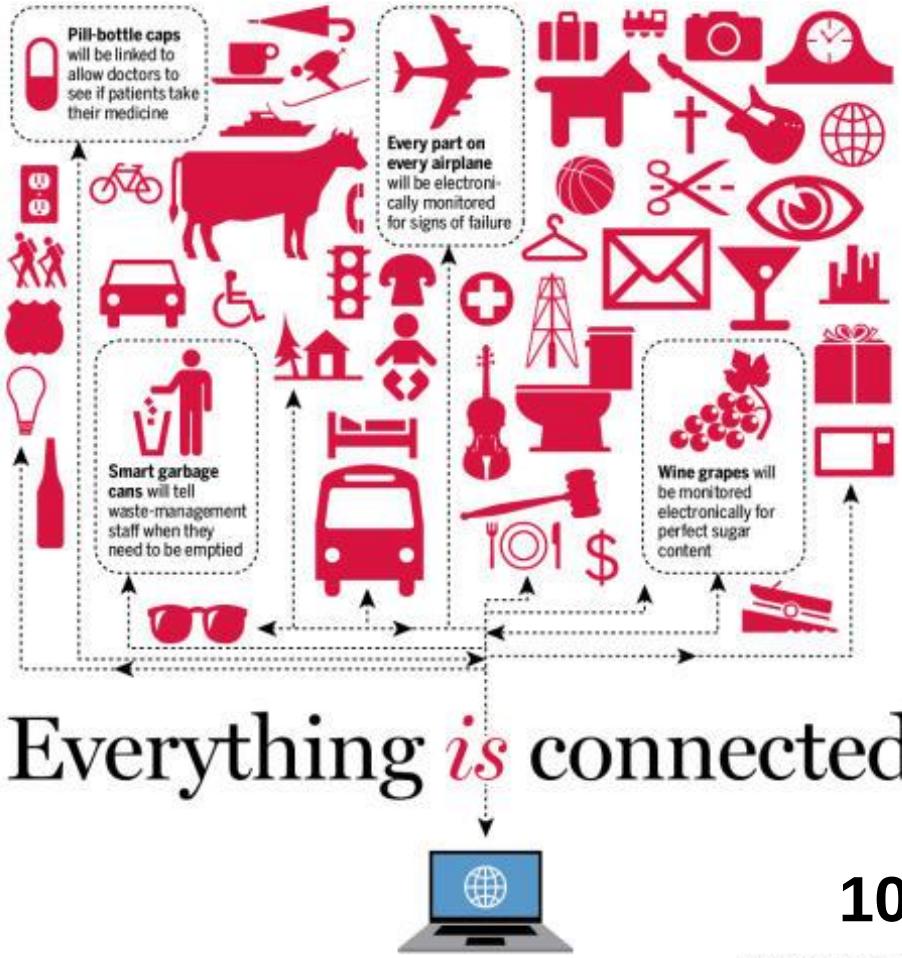
Evolution of Modern Computing Platforms

Computing Evolution – Internet of Things

- The term, "***Internet of Things***" was coined in 1999 by Kevin Ashton when he was working in the Media Center at the Massachusetts Institute of Technology.
- The Internet of Things (IoT) is an area of Information Technology that is exploding with promise and possibilities because of the rapid proliferation of inexpensive, yet powerful technologies both in hardware and in software. In fact, it has heralded a new paradigm shift in Internet-enabled computing, adding to and enhancing the present state of complex digital infrastructures.
- Still, IoT rapid adoption has also revealed its weaknesses in the areas of security, lack of privacy, and manageability.
- The Mirai Botnet Attack of October 2016 used known security weaknesses in tens of thousands of Internet of Things (IoT) Devices to launch massive Distributed Denial of Services Attacks against DYN, which is a major DNS Service provider. The result was a notable performance degrades in tens of thousands of businesses who rely heavily on the Internet

Computing Evolution – Internet of Things – Enabling Technologies

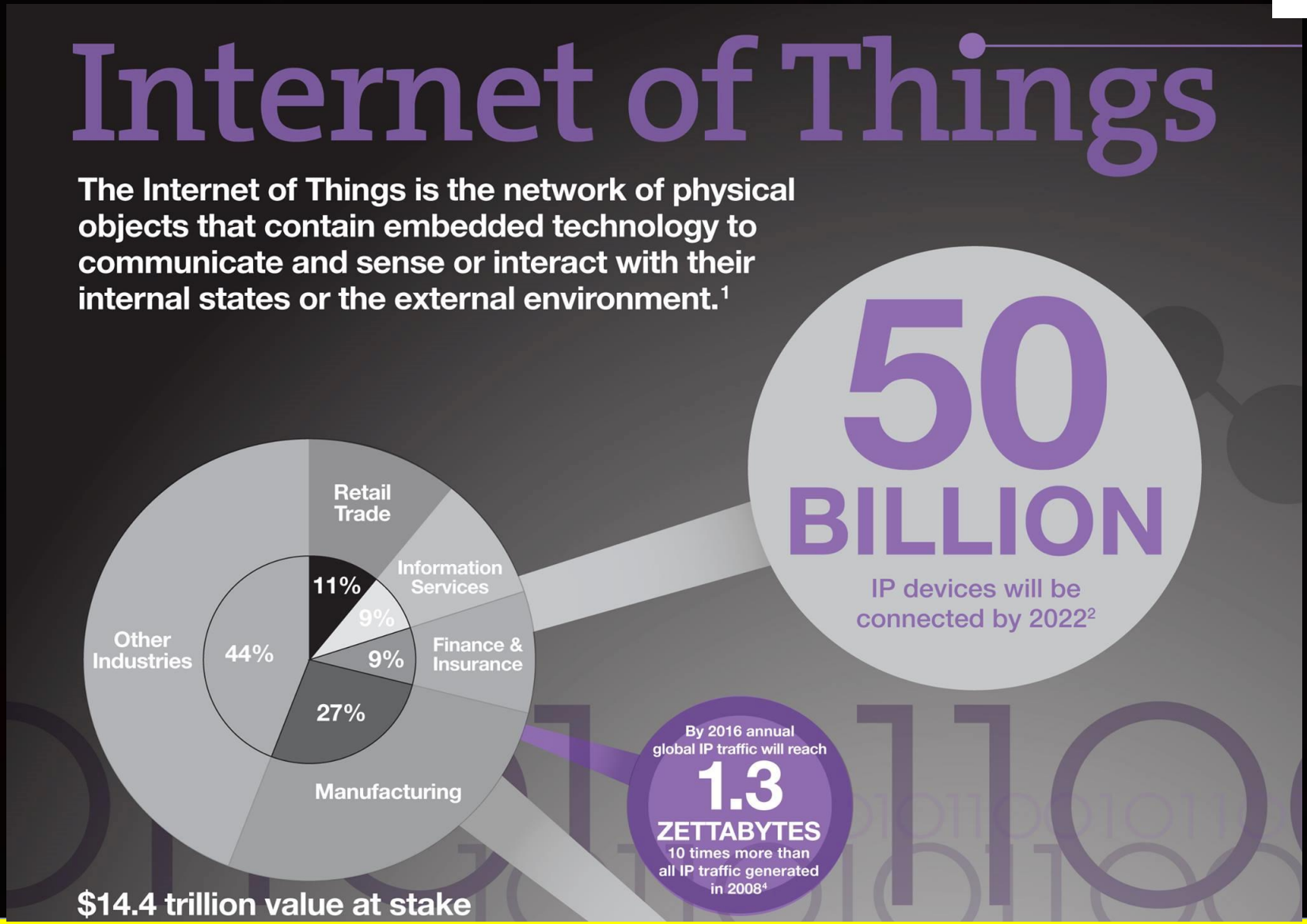
- The Internet
- The Cloud
- High bandwidth
- Web technologies
- Switching technologies
- High performance networking
- Lower cost, high-power CPUs
- Low-cost or no-cost operating systems
- Development tools
- Service Oriented Architecture
- Data Centers
- Network security
- Related management technologies



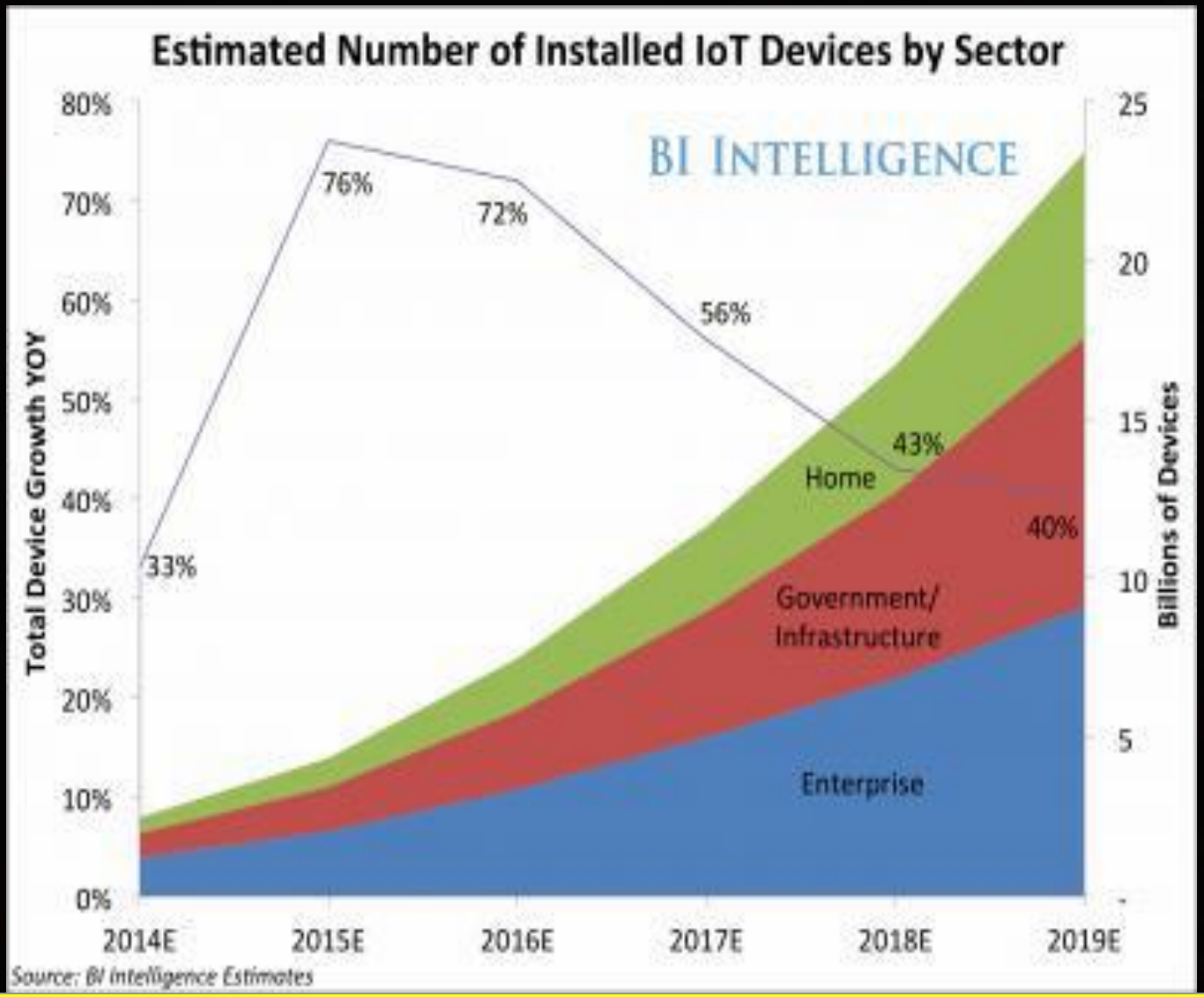
Computing Evolution – Internet of Things - Devices

- Raspberry Pi
- CCTV cameras
- DVRs
- Digital TVs
- Home routers
- Printers
- Alexa
- Security systems
- Garage doors
- Industrial systems
- Medical systems
- Home appliances
- Cars
- Other stuff

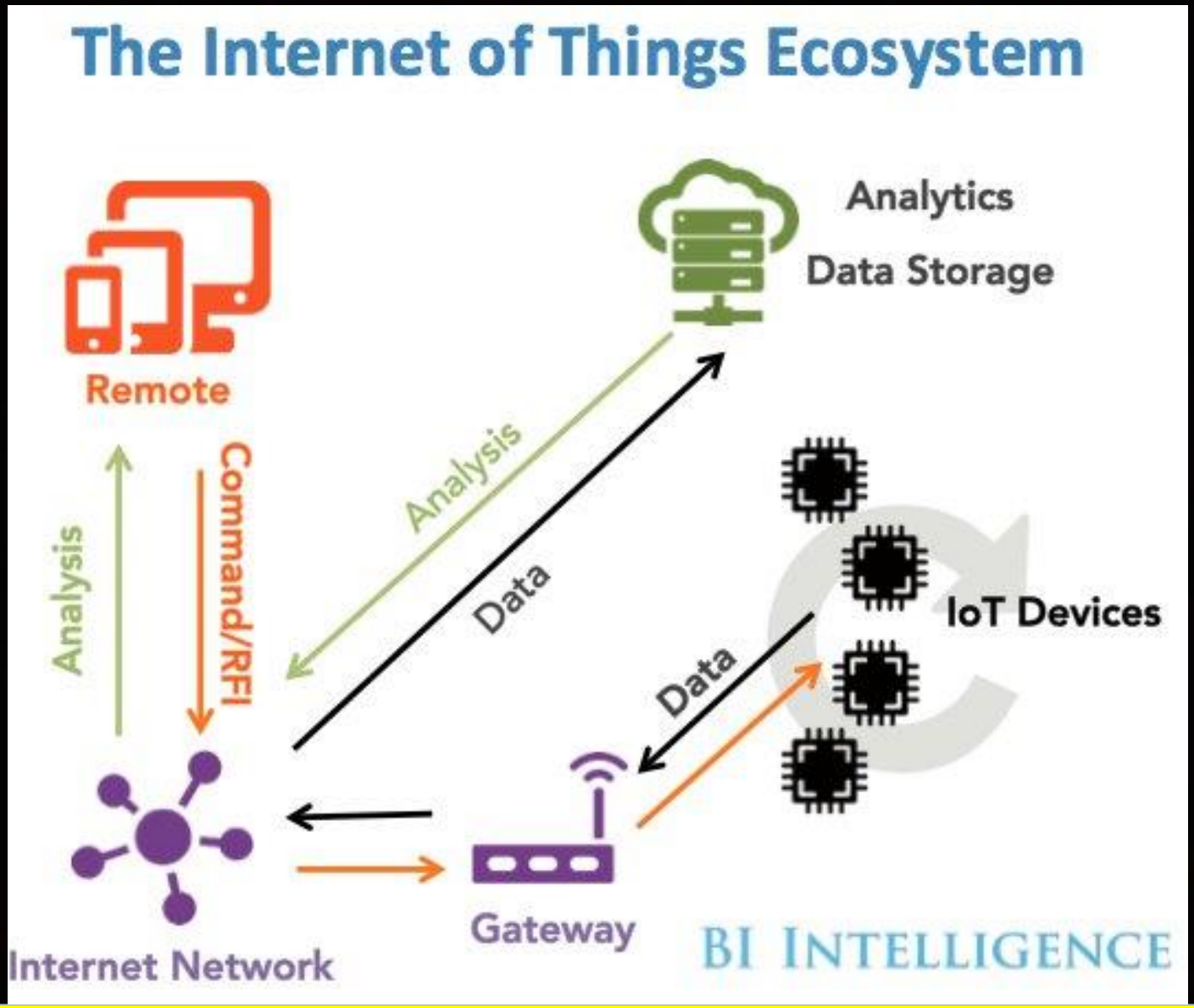
Computing Evolution – Internet of Things



Computing Evolution – Internet of Things



Computing Evolution – Internet of Things



Personal Computing, 1985 vs. 2021

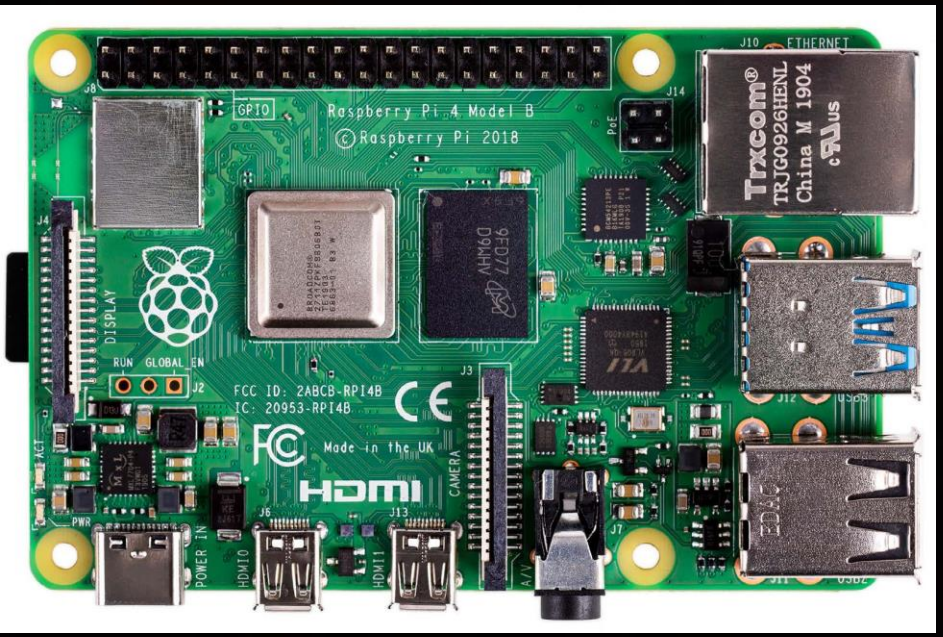
Personal Computing - March 1985

ITT Xtra PC – an IBM Clone PC
CPU: Intel i8088, 4.77 MHz
RAM: 256KB, 30 ns
Storage: 2 x 360 8.5” Floppy Drives
Bus: ISA
Ports: 2 RS-232 Serial, 1 Parallel
Video Card: VGA
Monitor: VGA
Modem: 1200 / 2400 Baud
Keyboard: QWERTY with Function Keys on Left
Operating System: MS DOS 2.11

TI 944 Dot-Matrix Printer
=========

Total Cost $4800

Personal Computing - August 2021



Raspberry Pi 4 B
CPU: Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
RAM: 8 GB
Storage: 2GB, 4GB or 8GB LPDDR4-3200 SDRAM
Bus: PCI Express
Ports: 2 USB
Video Card: 4-pole stereo audio and composite video port
H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
Network: 1 Gbs Ethernet
WiFi: 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Operating System: Debian Linux → Raspberry Pi OS
Keyboard: (Not Included) QWERTY

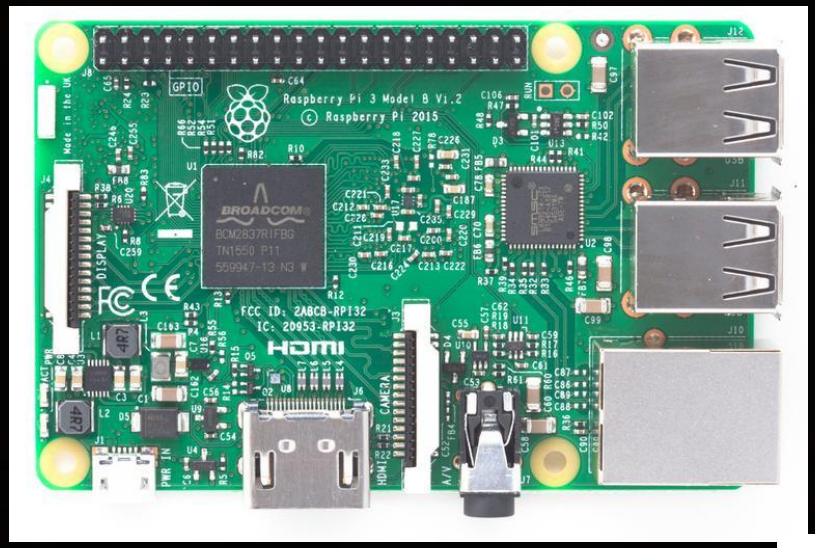
=========

Total Cost of a CANA KIT approximately $130.00

What is Raspberry Pi?

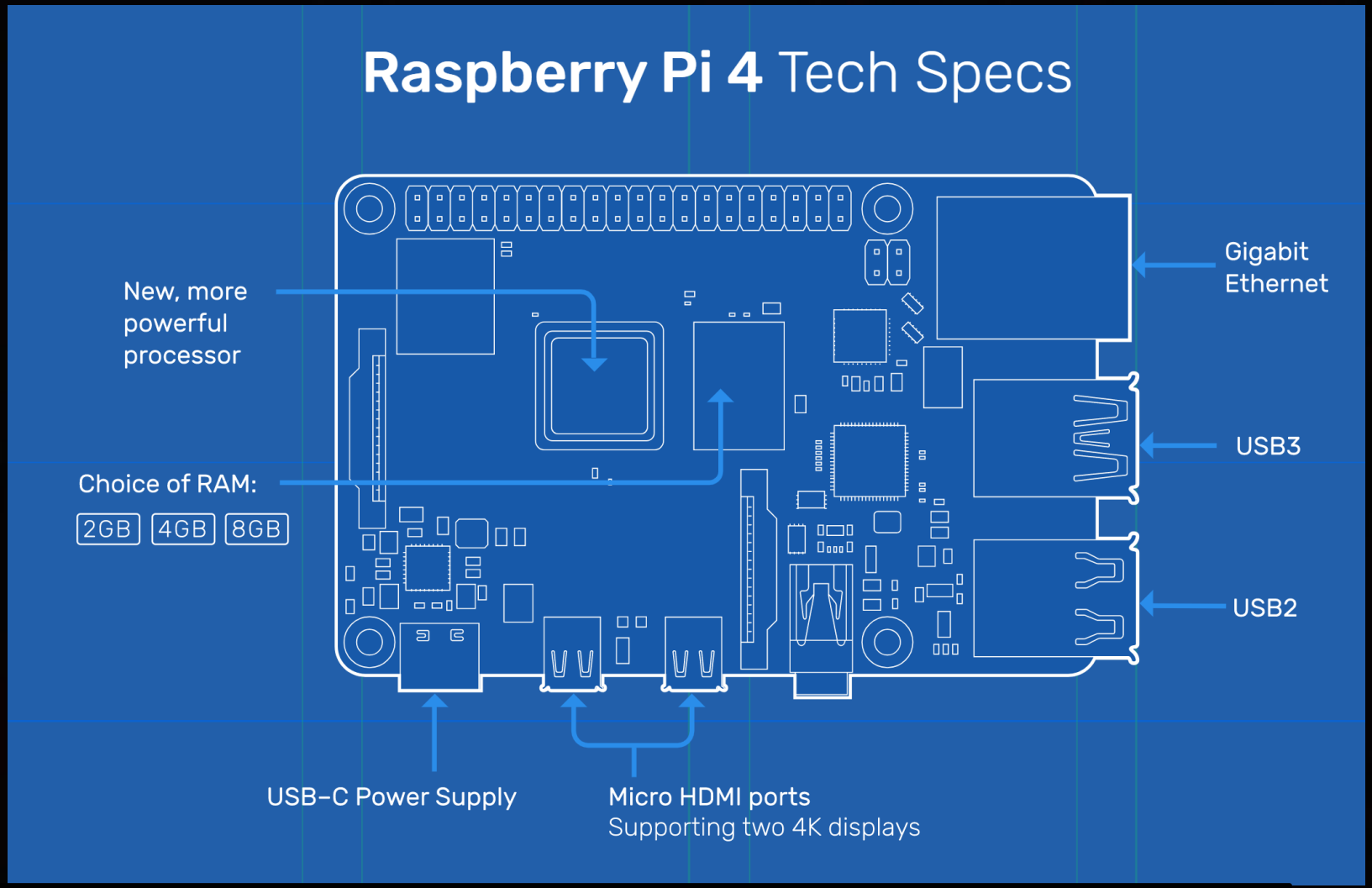
What is Raspberry Pi 4?

- A powerful, inexpensive ARM computer that runs Raspberry Pi OS
- Invented in 2010 by Eben Upton, Rob Mullins, Jack Lang, and Alan Mycroft at the University of Cambridge in the U.K.
- An Answer to the plight of technical computer illiteracy
- As of the end of 2019, over 30 million units shipped



Tung, L.(2019). https://www.zdnet.com/article/raspberry-pi-now-weve-sold-30-million/

Raspberry Pi 4 Specifications



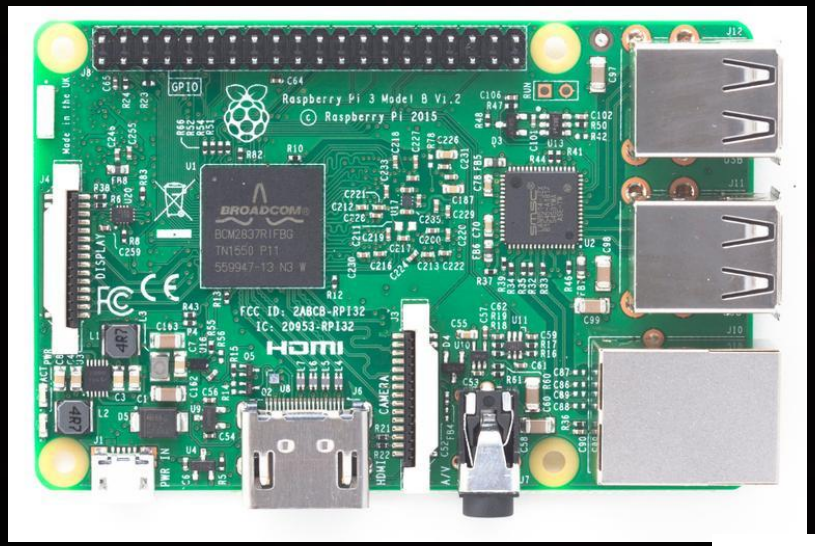
Raspberry Pi (2021). https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

Raspberry Pi 4 Specifications (2019)

Specifications

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient

* A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.



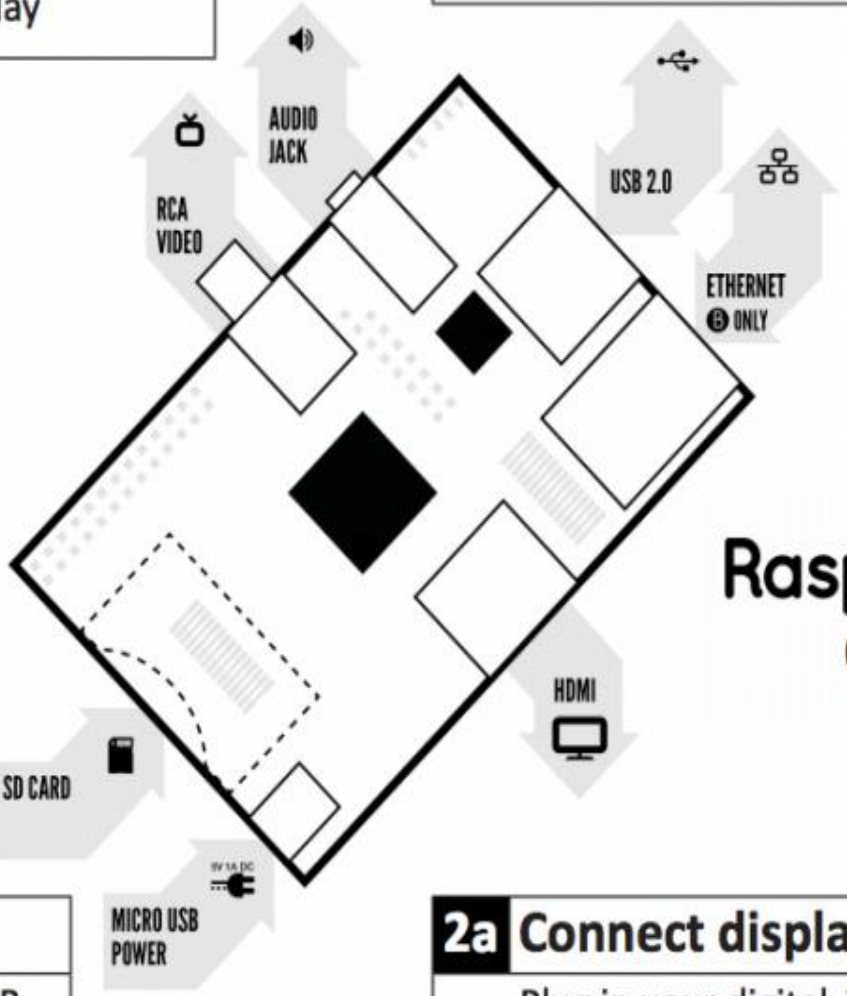
Raspberry Pi (2021). https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

2b Connect display

If *not* using HDMI, plug in your analogue TV or display

3 Connect input

Plug in a USB keyboard and mouse



4 Connect network

Connect to your wired network [optional]

Raspberry Pi Quick start

1 Insert SD card

See page 3 for how to prepare the SD card

5 Power up

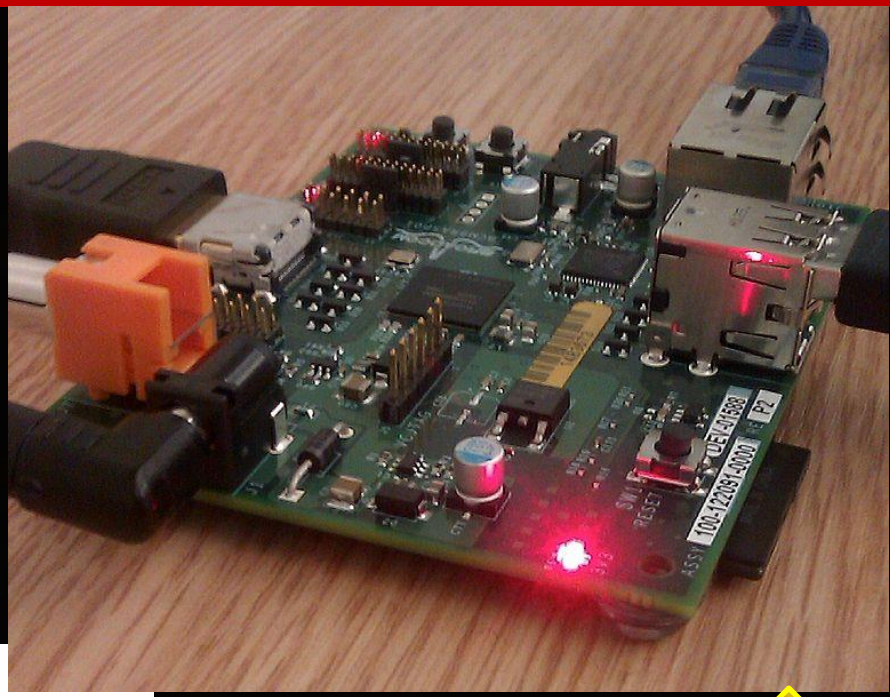
Plug in the micro USB power supply

2a Connect display

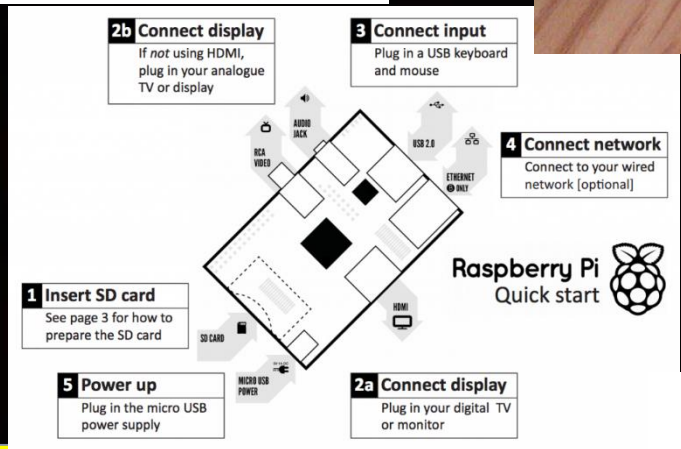
Plug in your digital TV or monitor

Raspberry Pi

Buy a Raspberry Pi CanaKit



Connect



Power up and Go!

Raspberry Pi 101
Getting Started

Raspberry Pi

- Buy a Raspberry Pi Kit – Preferably the Cana Kit
- Go to this URL and download the Getting Started with Raspberry Pi PDF file: https://goo.gl/bTb4mr
- It will easily step you through the process of starting up your new Raspberry Pi.

What is Ethereum?

What is Ethereum?

- A distributed, decentralized blockchain-based ledger
- Recently converted to a Proof-of-Stake Consensus protocol in August 2021
- Invented and released in 2015 by Vitalik Buterin, Ethereum is based on the Bitcoin Blockchain, except that it allows for programmable Smart Contracts
- Facilitates the trade of the Ether Cryptocurrency
- It uses the Ethereum Virtual machine as an execution environment
- Many feel the features of Ethereum environment will offer Blockchain application designers and developers to more fully realize the potential for Blockchain to change the world.
- Available for FREE on Github: https://github.com/ethereum
- Maintained by the Ethereum Foundation (www.ethereum.org)

Putting Ethereum on Raspberry Pi

Installing Raspberry Pi OS

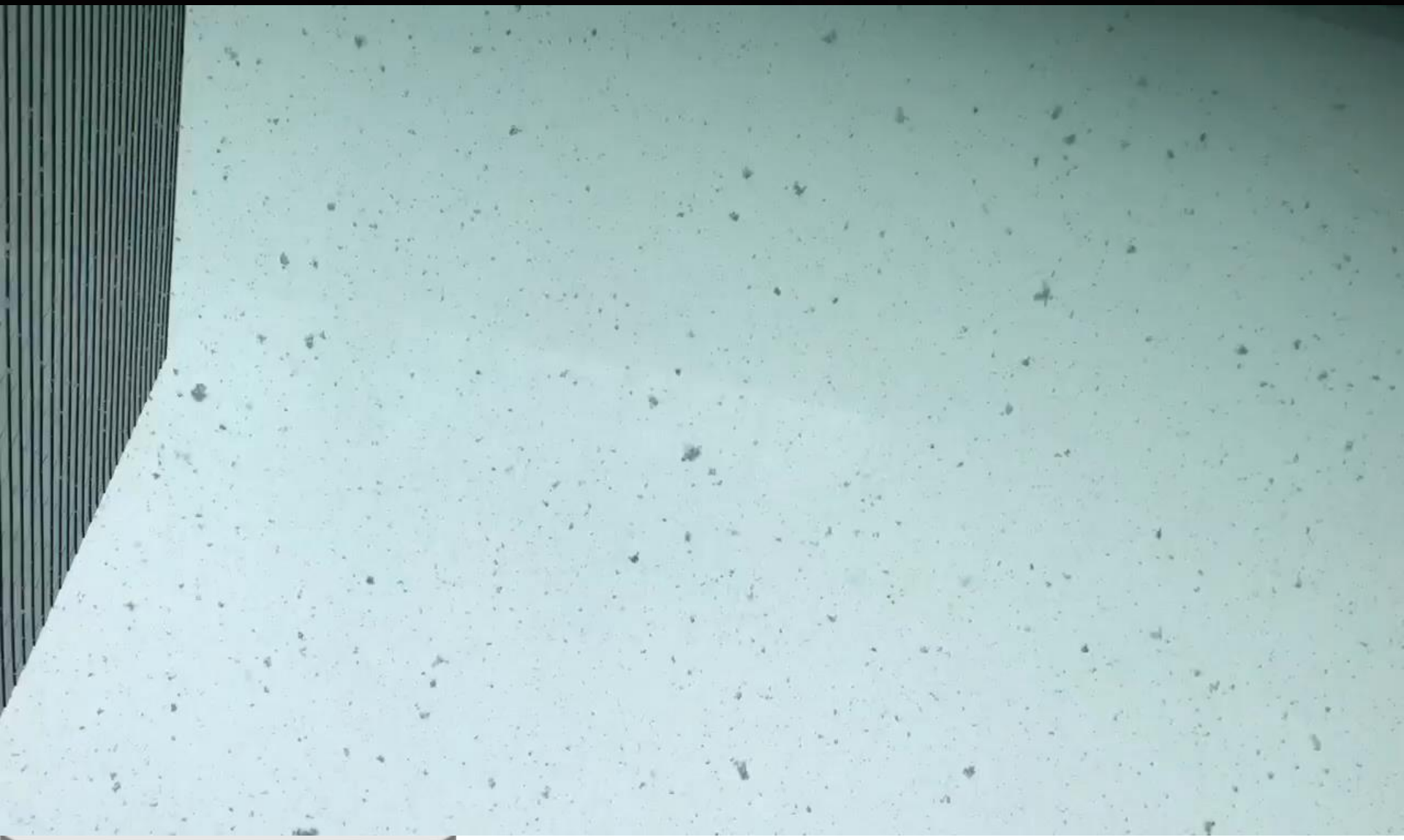
- Operating system for Raspberry Pi
- Running system requires Raspberry pi 4 (8gb RAM)
- Go to https://www.raspberrypi.org/software/
- Download RBP imager and updated package of Debian

Installing Ethereum

- Download Ethereum from GitHub
- Install Ethereum (Geth)
- Configure and Initialize Ethereum
- Configure and Initialize Ethereum Network
- Install Ethereum (Geth) on a Second Node
- Configure and Initialize Ethereum on a Second Node
- See Supplemental Slides for Additional Reference and Examples

What is Fluree?

What is Fluree?



Slater Technologies

PULSE CONNECT

What is Fluree?

Slater Technologies

PULSE CONNECT

What is Fluree?

Trust Your Data

3 key Fluree features that bring trust to data

1. 

Immutable & tamper proof
Blockchain hashing, transaction signatures

2. 

Embedded Data Permissions
SmartFunctions (permissions, “specs”, query/read)

3. 

Decentralization
Distribute transaction validation responsibilities

What is Fluree?

Leverage

3 key Fluree features that give your data more leverage

1 

Semantic Graph
Ideal "fit" for modern apps, AI. Link data without data warehouses.

2 

Time Travel
Freeze time across computation, troubleshoot issues, improve support

3 

Real-Time Applications
Fluree JavaScript Library, ledger "watch"

What is Fluree?

Fluree Data Stack

Apps or Services

- Deliver queries directly to apps via SmartFunctions
- Embed libraries directly in-app for in-memory speeds

Integration Protocols

- Interact with native standards
- GraphQL, SPARQL, FlureeQL

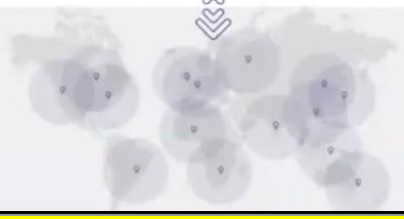
EDGE Network

- FlureeDB is a real-time content delivery network (CDN) that can be geographically dispersed
- Allows linear scale for *billions of queries per second with sub-millisecond latency*. Data may be leveraged by billions of end users.

RDF-based Immutable Blockchain Ledger

- FlureeDL is an immutable ledger of every state change formatted in RDF
- Partitioning across geographies for compliance/regulation

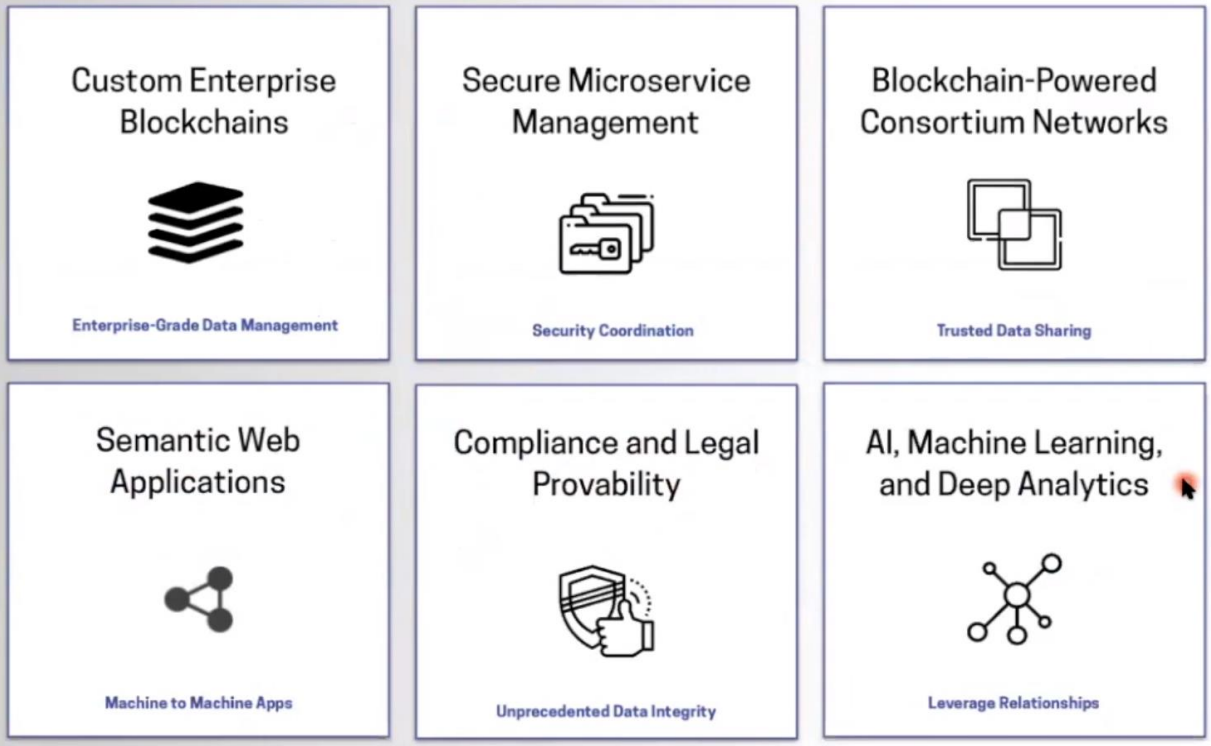
Exchange Consortium



- Optional Consortium Participants added to participate in transaction validation
- Raft or PBFT

What is Fluree?

Fluree fits **emerging data needs**.



Putting Fluree on Raspberry Pi

Installing Raspberry Pi OS

- Operating system for Raspberry Pi
- Running system requires Raspberry pi 4 (8gb RAM)
- Got to https://www.raspberrypi.org/software/
- Download RBP imager and updated package of Debian

Install Java JDK- version 11

- Java JDK 11(Debian package) is an open source utility that is needed in order to run Fluree.
- Register a new user account at www.oracle.com to download Java JDK.
- Open archiver, locate the JDK file and extract it
- Install file in the terminal

Install Fluree Instance

- Go to docs.flur.ee webpage and download the stable most recent version of Fluree.
- Next go to the archive on the RBP device locate the downloaded file and extract it.
- Once unzipped locate the file in the directory and call [./fluree_start.sh] to implement the Fluree start up file.

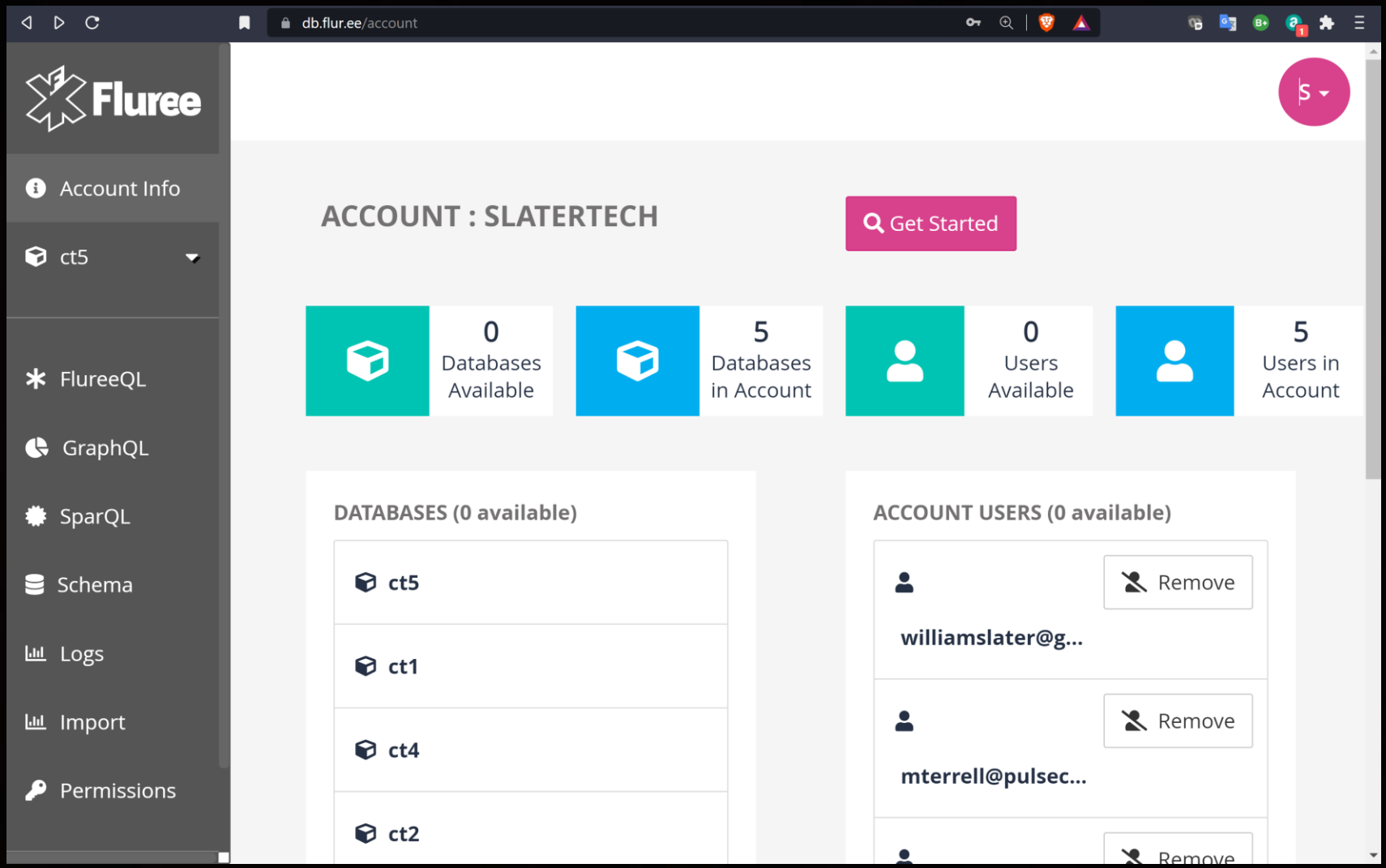
Install node.JS

- Find updated Version of Node.js for Fluree
- Install the zip file/ and extract it.

Starting / Run the ledger

- Open Raspberry Pi's terminal and call to Fluree db start up package [./fluree_start.sh]
- Open the web browser and go to [http://localhost8080]
- This browser will connect the Raspberry Pi to the local host ledger.

How Fluree Appears in a Web Browser



Fluree on Raspberry Pi -- More Information

If you are interested in participating and being one of the first countries to have access to our Decentralized HealthHub application, please contact Randolph Rodriguez, CEO of Pulse Connect at rrodriguez@pulseconnect.net or 1 (224) 288 7066

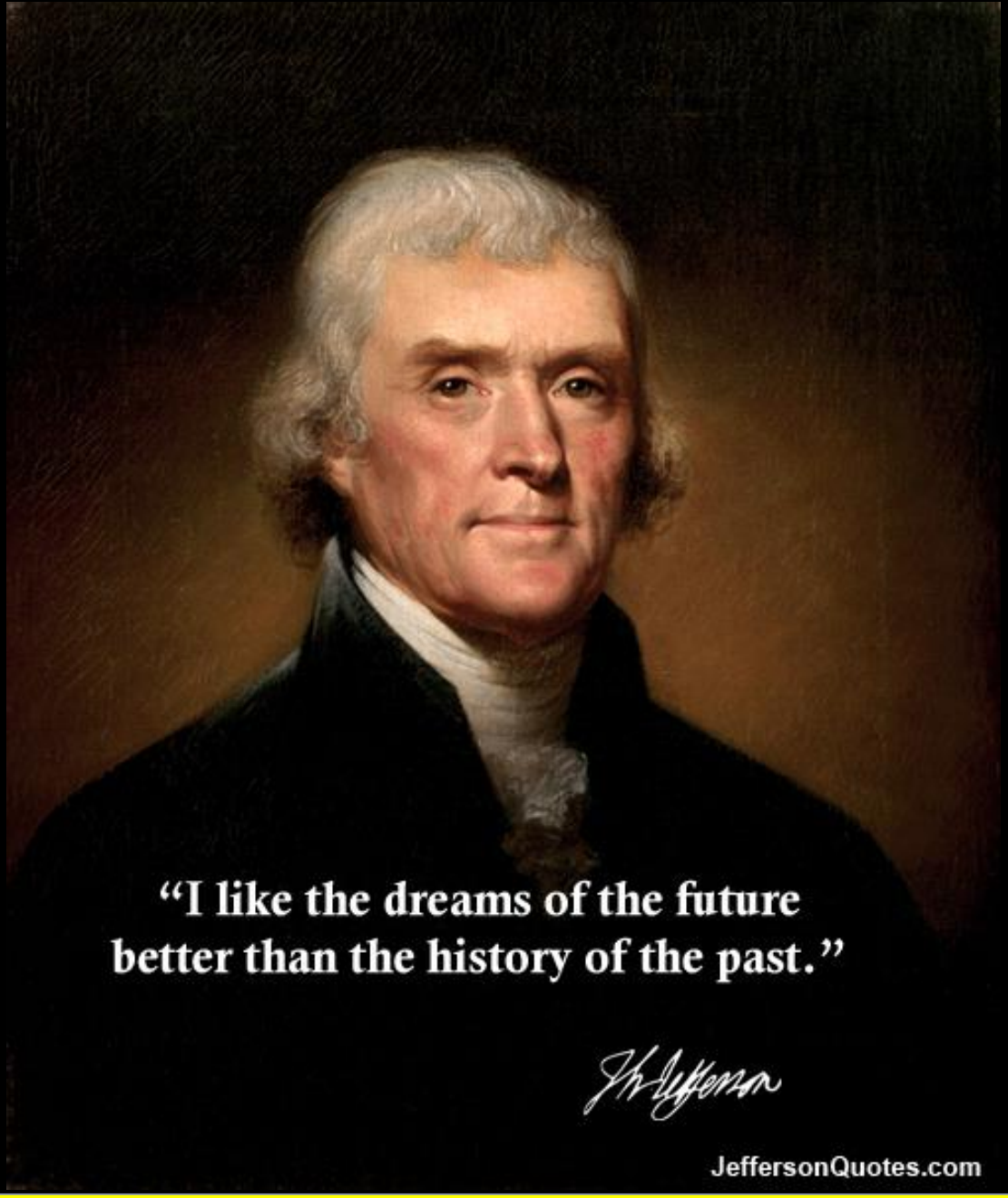
If you are interested in in working with the Fluree Platform please contact Buck Flannigan, VP of Fluree Marketing and Partner Management at buck@flur.ee

Conclusion

Conclusion

- We covered:
  - Computing Evolution
  - Internet of Things
  - Raspberry Pi
  - Ethereum
  - Fluree

Parting Thoughts


"I like the dreams of the future
better than the history of the past."
JeffersonQuotes.com

I have learned that people will forget what you said, people will forget what you did, but people will never forget how you made them feel.

Maya Angelou
1928-2014

Parting Thoughts:
Like Records on a Blockchain, Let Our Love, Support, & Friendship Be Immutable & Enduring

Presenter Bio: William Favre Slater, III

- **Project Manager / Sr. IT Consultant at Slater Technologies, Inc. , and Adjunct Professor at the Illinois Institute of Technology -** Working on projects related to:
  - Security reviews and auditing
  - ISO 27001 Project Implementations
  - Developing Applications for Risk and Compliance
  - Subject Matter Expert for Government Proposals and Contracts related to technical services management and measurement
  - SME for preparing Risk Management and Security Exams at Western Governor's State University in UT
  - Created an eBook with articles about Security, Risk Management, Cyberwarfare, Project Management and Data Center Operations
  - Providing subject matter expert services to Data Center product vendors and other local businesses.
  - Developing and presenting technical training materials for undergraduate and graduate students at the ***Illinois Institute of Technology*** in the areas of Data Center Operations, Data Center Architecture, Cyber Security Management, and Information Technology hardware and software.
  - Providing Summer Internships to IIT Students via his company, Slater Technologies, Inc.
  - Doing Internet of Things Projects


Pushing the Raspberry Pi:
Ethereum on Pi and Fluree on Pi.
Low-cost and Open Source
Enterprise Computing for Fun,
Learning, and Training.
Prepared for the
2021 CAPDA ICT Africa
SYMPOSIUM
Conference in
Yaounde, Cameroon,
Central Africa
Thank You!
Questions & Answers

William Favre Slater, III

- President / CEO / CISO of
  Slater Tecchnologies, Inc
- 312-758-0307
- slater@billslater.com
- williamslater@gmail.com
- http://billslater.com/interview
- 1515 W. Haddon Ave., Unit 309
  Chicago, IL 60642
  United States of America

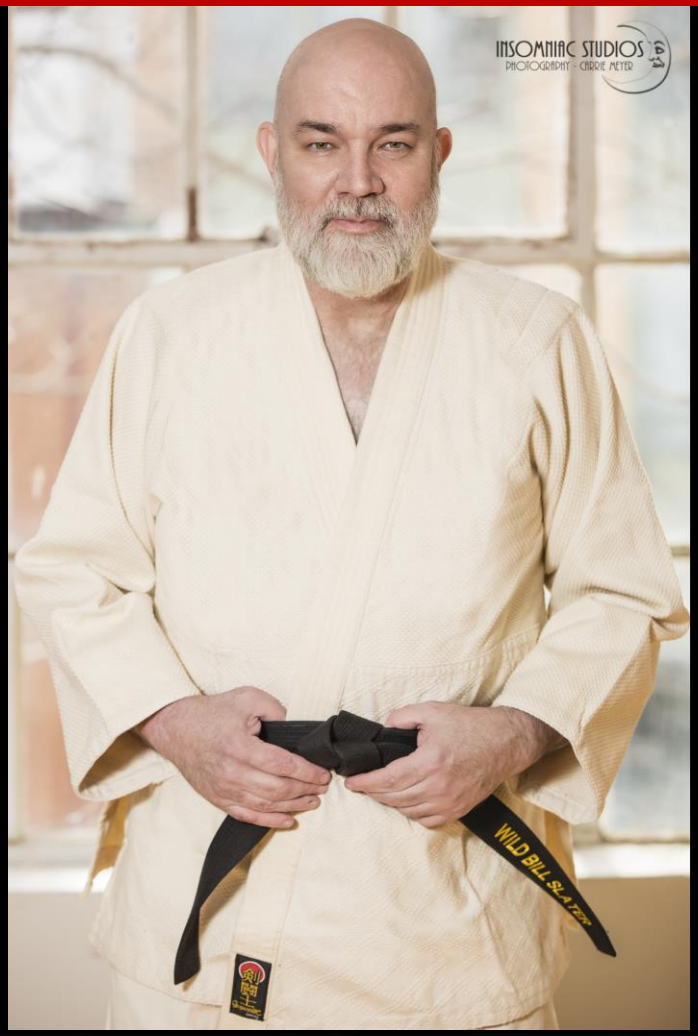
William Favre Slater, III

Slater Technologies

PULSE CONNECT

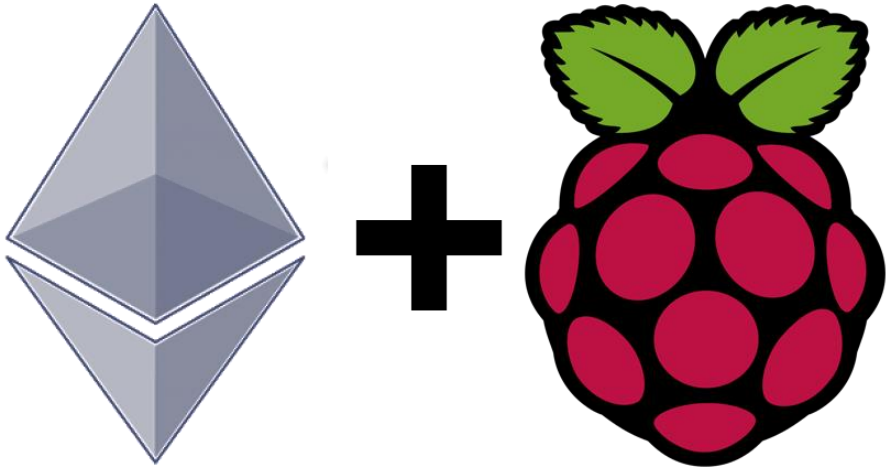
Questions

Thank You!

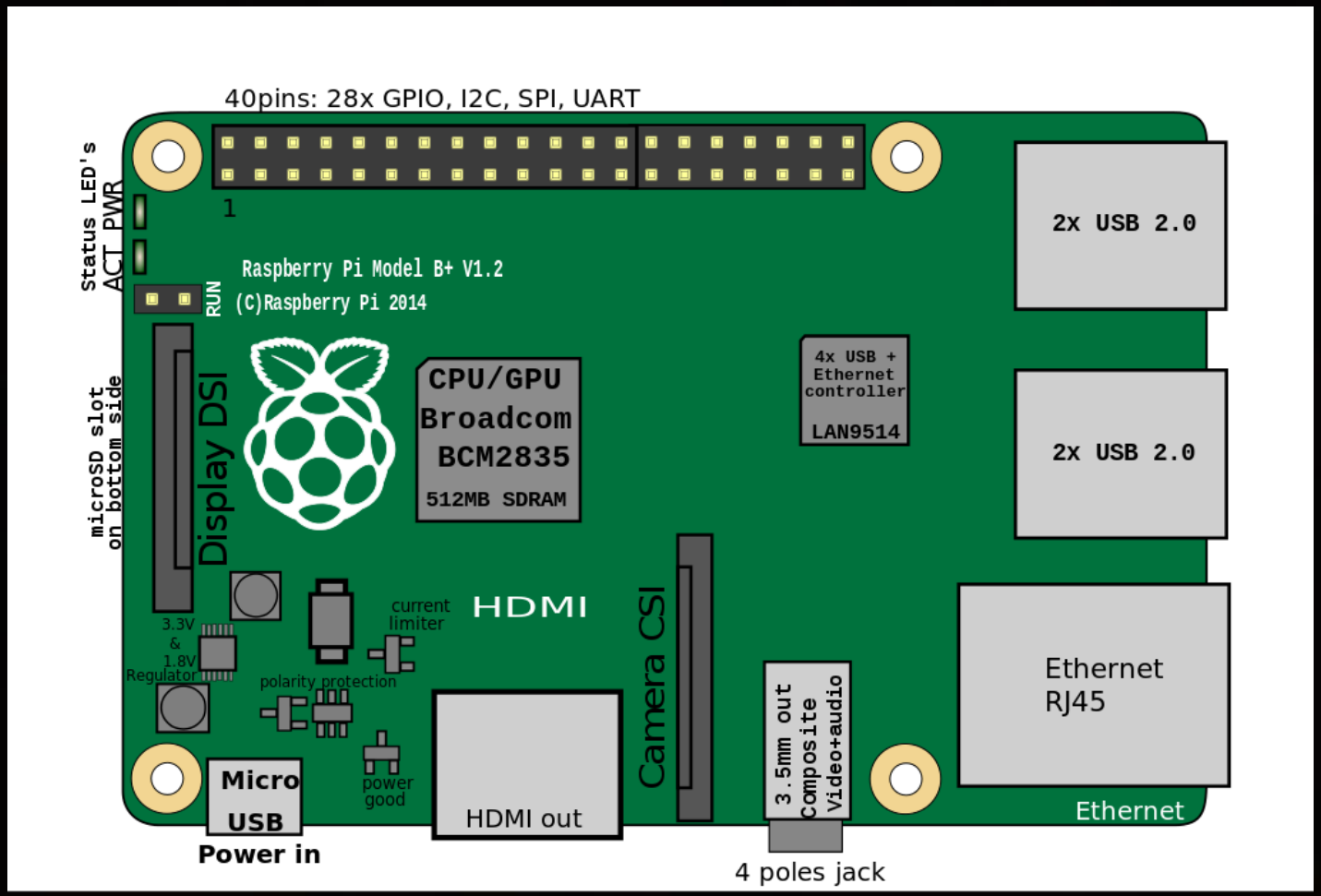

INSOMNIAC STUDIOS
PHOTOGRAPHY - CARRIE MEYER
WILD BILL SLATER

Supplemental Slides

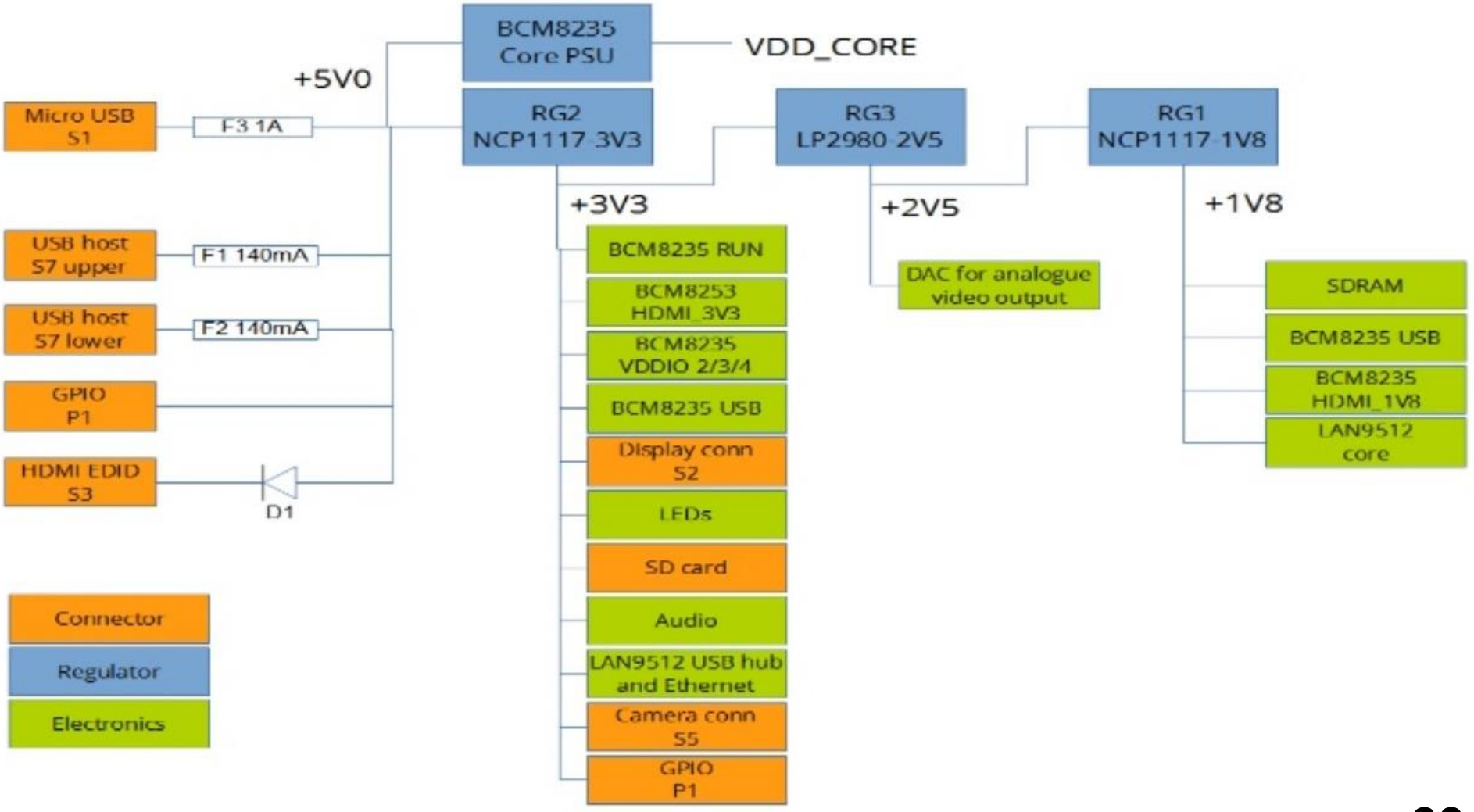
Introduction to Setting Up Ethereum on a Small Raspberry Pi Network



Raspberry Pi Architecture



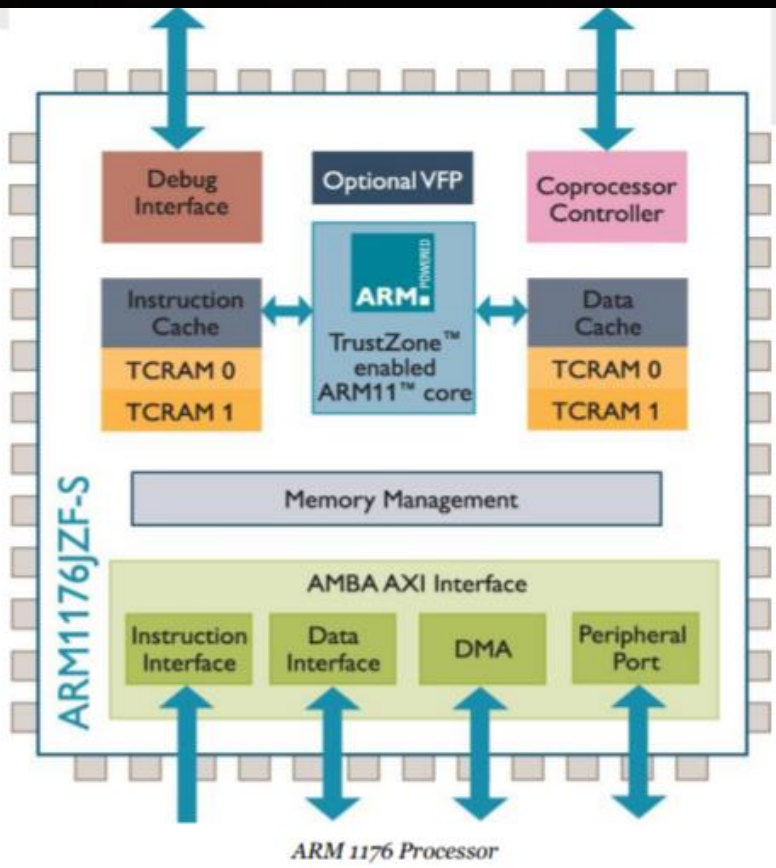
Raspberry Pi Architecture



Fazal, R. (2014). Raspberry Pi.

Broadcom BCM 2835 Architecture

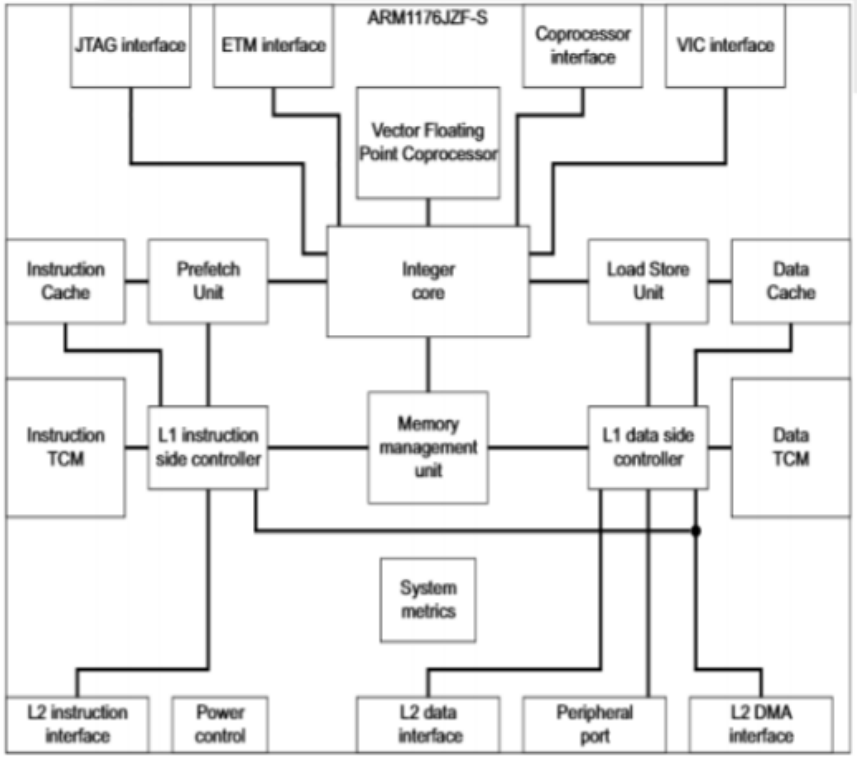
- ARM11J6JZF-S (ARM11 Family)
- ARMv6 Architecture
- Single Core
- 32-Bit RISC
- 700 MHz Clock Rate
- 8 Pipeline Stages
- Branch Prediction



Holton, J. and Fratangelo, T. (2016). Raspberry Pi Architecture.

Broadcom BCM 2835 Overview with Block Diagram

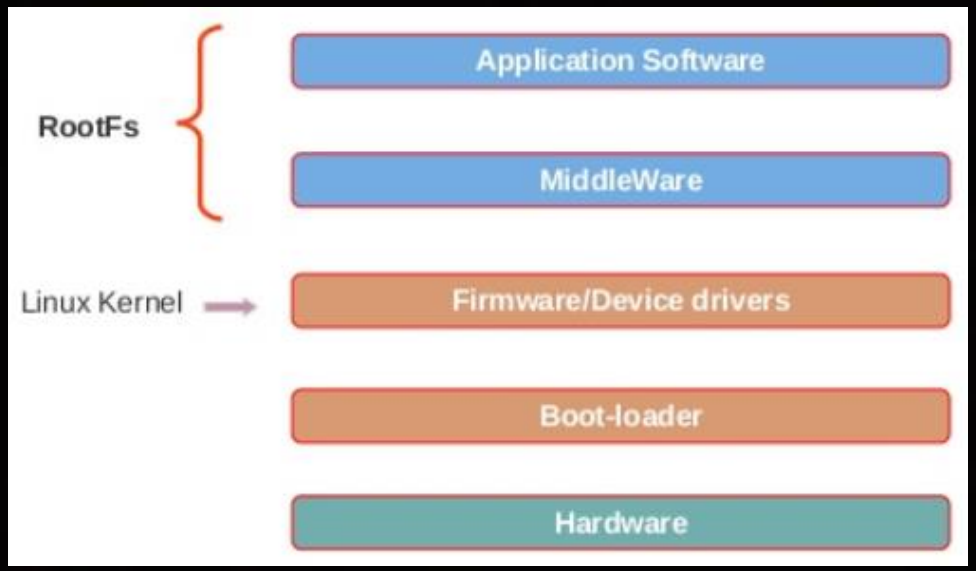
- Core
- Load Store Unit
- Prefetch Unit
- Memory System
- Level One Mem. System
- Interrupt Handling
- System Control
- AMBA Interface
- Coprocessor Interface
- Debug
- Instruction cycle summary and interlocks
- Vector Floating-Point



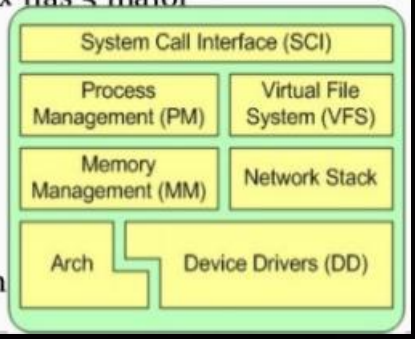
ARM1176JZF-S Technical Reference Manual

Holton, J. and Fratangelo, T. (2016). Raspberry Pi Architecture.

Raspian Linux Architecture



- Within the kernel layer Linux has 5 major subsystems.
  - Process Scheduler
  - Memory Manager
  - Virtual File System
  - Network interface
  - Inter process communication



When You Install and Start Up Raspberry Pi OS

2.6 Raspbian's Desktop Environment

If you have not changed the setting in raspi-config, the Raspberry Pi will boot into Raspbian's command line.

To start the desktop environment:

1. Type *pi* as the username, then press **Enter**.
2. Type your password, then press **Enter**.[1]
3. Type the following command and press **Enter**:

```
startx
```

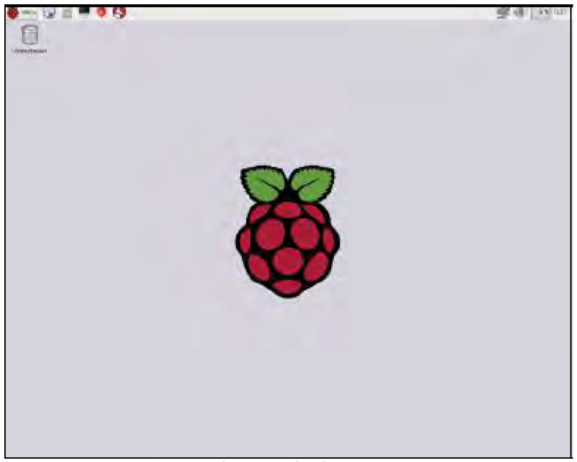
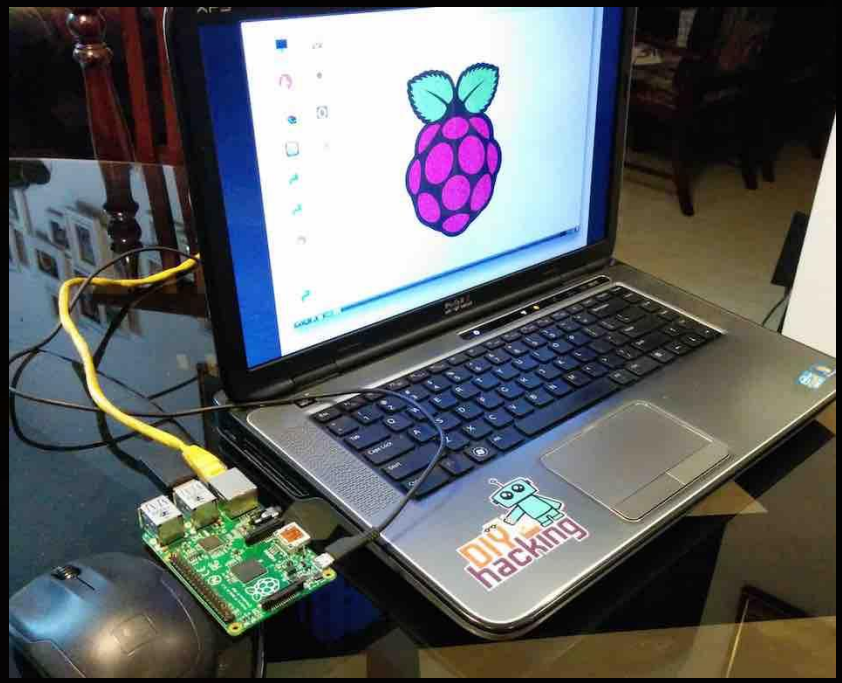


Figure 2. Raspbian's desktop environment

Using Your Windows Laptop Display with Raspberry Pi

Follow the excellent tutorial and it will show you
how to connect your Raspberry Pi to your
Windows Laptop.

https://maker.pro/raspberry-pi/tutorial/how-to-connect-a-raspberry-pi-to-a-laptop-display



Note to Mac users (and I know there are multitudes out there) I don't do Mac, only Windows and Linux – Sorry.

Patel, S. (2015). Connecting Your Raspberry Pi to a Laptop Display

SOME TERMS

Some Important Terms

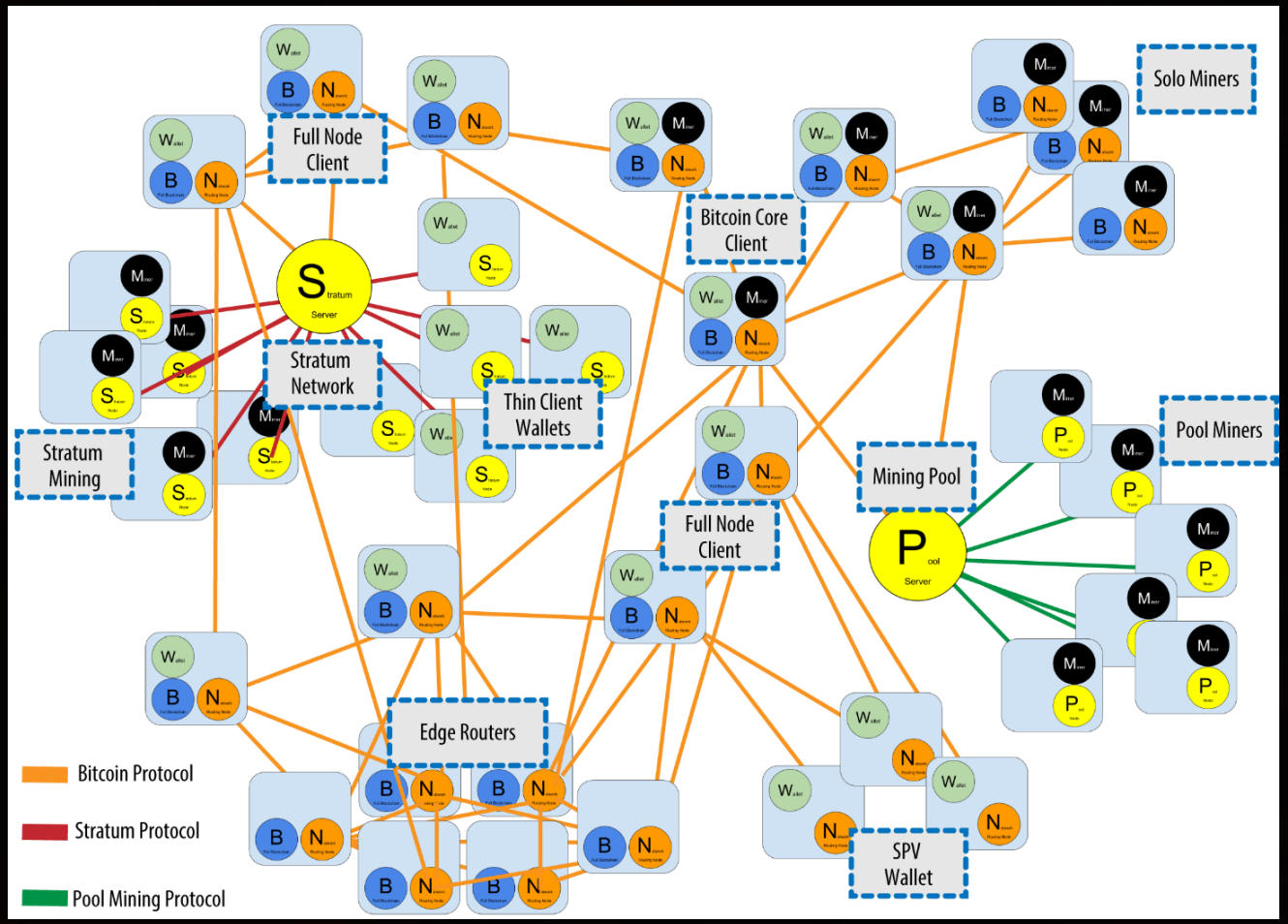
Term	Explanation
AES SHA-256	The 256-bit encryption algorithm that is AES standard used for Bitcoin keys.
Bitcoin Network	The Internet-connected network comprised of the software and data that supports Bitcoin transactioms
Blockchain	The Bitcoin ledger of past transactions.
Difficulty	The measure of how difficult it is to find a new block compared to the easiest it can ever be
Exchange	A place that sells can buys Bitcoins, like a stock exchange.
Hash	It is a standard cryptographic algorithm function for the generation and verification of currency
Mining	Bitcoin mining serves 2 purposes, it creates the general ledger of Bitcoin transactions and it provides security.
Private Key	The secret cryptographic key that is used to protect your Bitcoin account
Proof of Work	An economic time-stamped measure to deter service abuses on a network by requiring some work from the service requester, usually meaning processing time by a computer.
Public Key	The public (shared) cryptographic key that is used to protect your Bitcoin account
Transaction	Use of the Bitcoin to purchase good or services, or the purchase of sale of a Bitcoin, or fractional part of Bitcoin
Wallet	A service that will safely store your Bitcoin account for you.

- *Candidate block*: An incomplete block, created as a temporary construct by a miner to store transactions from the transaction pool. It becomes a complete block after the header is completed by solving the PoW problem.
- *PoW* : The problem of discovering a new hash that can be used in the block header of the candidate block. This is a computationally intensive process that involves evaluating a hash taken from the most recent block and appending a nonce to it against the target value of the network. This problem can only be solved using brute force; that is, multiple trials of using the hash (from the most recent block header) and nonce being adjusted each time are necessary to solve the PoW problem.
- *Nonce*: A 32-bit value that is concatenated to the hash from the most recent block header. This value is continuously updated and adjusted for each trial, until a new hash below the target value is discovered.
- *Hash function*: A function used to compute a hash. In the Bitcoin protocol, this function is the SHA-256.
- *Hash value*: The resulting hash output from a hash function.
- *Target value*: A 265-bit number that all Bitcoin clients share. It is determined by the difficulty, which is discussed shortly.
- *Coinbase transaction*: The first transaction that is packaged into a block. This is a reward for the miner to mine the PoW solution for the candidate block.
- *Block header*: The header of a block, which contains many features such as a timestamp, PoW, and more. We describe the block header in more detail in Chapter 3.

Source: **Blockchain Basics: A Non-technical Introduction in 25 Steps by Daniel Drescher**

WHAT IS BLOCKCHAIN?

A Logical Diagram of a Blockchain Network



What Is Blockchain?

- Distributed Ledger
- Decentralized
- Popularized by Satoshi Nakamoto
- Uses Cryptography and Hashing
- Append-only Transactions
- The Code already exists in Github
- Immutable
- First discussed in 1991
- Ethereum announced in 2015

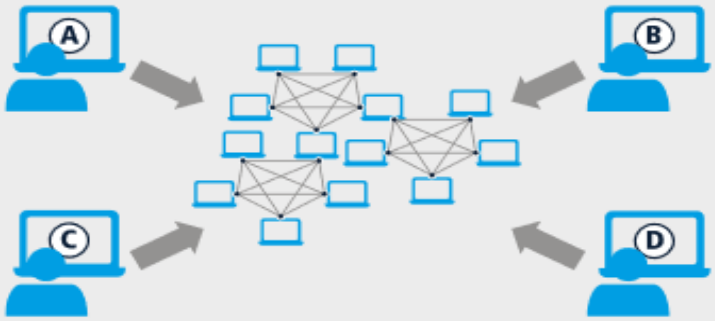
Ethereum Public Blockchain

- Ethereum was developed initially for public chain deployment, where trustless transaction requirements outweigh absolute performance. The current public chain consensus algorithms (notably PoW) are overkill for networks with trusted actors and high throughput requirements.
- Public chains by definition have limited (at least initially) privacy and permissioning requirements. Although Ethereum does enable permissioning to be implemented within the smart contract and network layers, it is not readily compatible out of the box with traditional enterprise security and identity architectures or data privacy requirements.
- Naturally, the current Ethereum improvement process (dominated by Ethereum improvement proposals) is largely dominated by public chain matters, and it has been previously challenging for enterprise IT requirements to be clarified and prioritized within it.

Source: **Blockchain Basics: A Non-technical Introduction in 25 Steps**
by Daniel Drescher

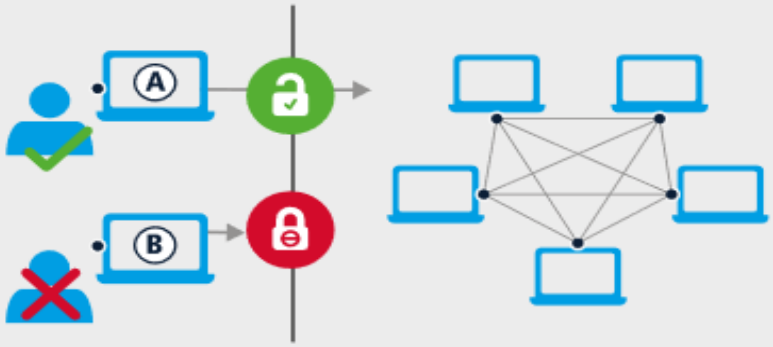
Public vs. Private

PUBLIC VS. PRIVATE BLOCKCHAINS



PUBLIC, PERMISSIONLESS BLOCKCHAINS

- Anyone can join the network and submit transactions
- Anyone can contribute computing power to the network and broadcast network data
- All transactions are broadcast publicly



PRIVATE, PERMISSIONED BLOCKCHAINS

- Only safelisted (checked) participants can join the network
- Only safelisted (checked) participants can contribute computing power to the network and broadcast network data
- Access privileges determine the extent to which each safelisted participant can contribute data to the network and access data from the network

Key differences between public, permissionless blockchains and private, permissioned blockchains; **Source:** Accenture

Four Functional Versions of Blockchain Distributed Ledgers

Table 23-2 presents the four versions of the blockchain that arise when combining the extreme cases of reading and writing restrictions.

Table 23-2. Four Versions of the Blockchain as a Result of Combining Reading and Writing Restrictions

	Reading Access and Creation of Transactions	
Writing Access	**Everyone**	**Restricted**
Everyone	Public & Permissionless	Private & Permissionless
Restricted	Public & Permissioned	Private & Permissioned

Source: **Blockchain Basics: A Non-technical Introduction in 25 Steps** by Daniel Drescher

```
contract mortal {
    /* Define variable owner of the type address */
    address owner;

    /* This function is executed at initialization and sets the owner
    function mortal() { owner = msg.sender; }

    /* Function to recover the funds on the contract */
    function kill() { if (msg.sender == owner) selfdestruct(owner); }
}

contract greeter is mortal {
    /* Define variable greeting of the type string */
    string greeting;

    /* This runs when the contract is executed */
    function greeter(string _greeting) public {
        greeting = _greeting;
    }

    /* Main function */
    function greet() constant returns (string) {
        return greeting;
    }
}
```

An example Ethereum smart contract. Source: ethereum.org.

A Smart Contract Written in Solidity

INSTALLING ETHEREUM ON RASPBERRY PI

Installing Geth

Cloning into go-ethereum:

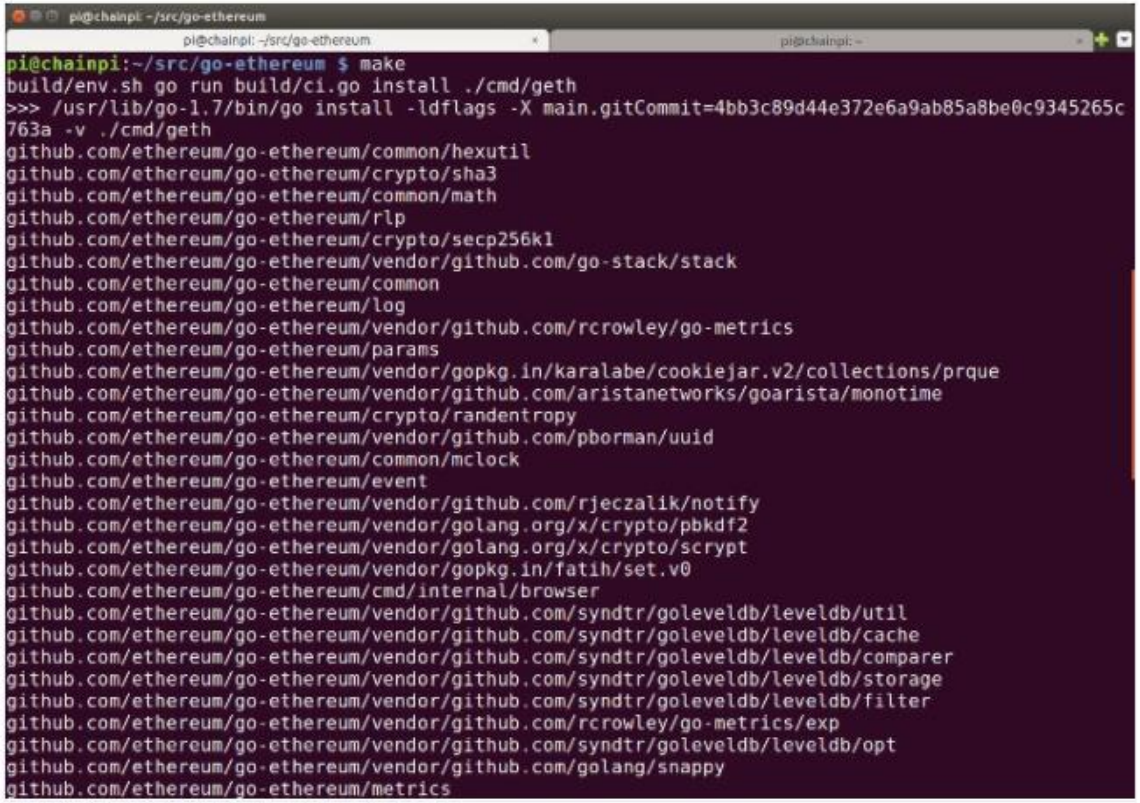
$ mkdir src

$ cd src

$ git clone -b release/1.7 https://github.com/ethereum/go-ethereum.git

```
pi@raspberrypi:~ $ mkdir src
pi@raspberrypi:~ $ cd src
pi@raspberrypi:~/src $ git clone -b release/1.7 https://github.com/ethereum/go-ethereum.git
Cloning into 'go-ethereum'...
remote: Counting objects: 69714, done.
remote: Total 69714 (delta 0), reused 0 (delta 0), pack-reused 69713
Receiving objects: 100% (69714/69714), 95.44 MiB | 248.00 KiB/s, done.
Resolving deltas: 100% (46419/46419), done.
```

Installing geth



Assuming we have already installed Raspbian, if we start by updating the installed packaged software to the latest versions.

Installing Geth

Back, A. (2017). Exploring Ethereum with Raspberry Pi - Part 1: Getting Started.

Start the Node

```
pi@chainpi:~ $
pi@chainpi:~ $ geth --syncmode light --cache 64 --maxpeers 12
INFO [01-30|17:38:56] Starting peer-to-peer node               instance=Geth/v1.7.3-stable-4bb3c89d/linux
-arm/go1.7.4
INFO [01-30|17:38:56] Allocated cache and file handles         database=/home/pi/.ethereum/geth/lightchai
ndata cache=64 handles=1024
INFO [01-30|17:38:56] Writing default main-net genesis block
INFO [01-30|17:39:02] Initialised chain configuration          config="{ChainID: 1 Homestead: 1150000 DAO
: 1920000 DAOSupport: true EIP150: 2463000 EIP155: 2675000 EIP158: 2675000 Byzantium: 4370000 Engine: eth
ash}"
INFO [01-30|17:39:02] Disk storage enabled for ethash caches   dir=/home/pi/.ethereum/geth/ethash count=3
INFO [01-30|17:39:02] Disk storage enabled for ethash DAGs     dir=/home/pi/.ethash               count=2
INFO [01-30|17:39:02] Added trusted checkpoint                 chain name="ETH mainnet"
INFO [01-30|17:39:02] Loaded most recent local header          number=0 hash=d4e567…cb8fa3 td=17179869184
INFO [01-30|17:39:02] Starting P2P networking
INFO [01-30|17:39:04] UDP listener up                          self=enode://b7a599e0eee28d102bed0e874e9b0
d76fe89b0b6fb06354f47339620c6010df4a8f4d5ec6092ef914e220d7a2e567530708be138faf6c2168fc86abdb818e52e@[::]:
30303
INFO [01-30|17:39:04] RLPx listener up                         self=enode://b7a599e0eee28d102bed0e874e9b0
d76fe89b0b6fb06354f47339620c6010df4a8f4d5ec6092ef914e220d7a2e567530708be138faf6c2168fc86abdb818e52e@[::]:
30303
WARN [01-30|17:39:04] Light client mode is an experimental feature
INFO [01-30|17:39:04] IPC endpoint opened: /home/pi/.ethereum/geth.ipc
```

```
$ geth --syncmode light --cache 64 --maxpeers 12
```


Back, A. (2017). Exploring Ethereum with Raspberry Pi - Part 1: Getting Started.

Start the Node

If we ran geth without any arguments, it would start up a node and attempt to sync the entire public mainnet blockchain. Which, at >50GB in size and constantly growing, might not be a great idea on an embedded computer. So instead we start the node in light synchronisation mode. This only fetches block headers as they appear and other parts of the blockchain on-demand.

To force the node to exit simply press CTRL-C. To run it as a service at boot time:

```
$ sudo vi /etc/systemd/system/geth@.service
```


Back, A. (2017). Exploring Ethereum with Raspberry Pi - Part 1: Getting Started.

Start the Node

(replace "vi" with your favourite text editor)

And then enter:

```
[Unit]
Description=Ethereum daemon
Requires=network.target

[Service]
Type=simple
User=%I
ExecStart=/usr/local/bin/geth --syncmode light --cache 64 --maxpeers 1
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Save the file. Following which to have the Ethereum node run as the “pi” user:

```
$ sudo systemctl enable geth@pi.service
$ sudo systemctl start geth@pi.service
```


Back, A. (2017). Exploring Ethereum with Raspberry Pi - Part 1: Getting Started.

Start the Node

```
pi@chainpi:~ $
pi@chainpi:~ $ geth attach
Welcome to the Geth JavaScript console!

instance: Geth/v1.7.3-stable-4bb3c89d/linux-arm/go1.7.4
 modules: admin:1.0 debug:1.0 eth:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> eth.accounts
["0xc0dad8541fd851d5094b4574899ebcf236cd3666"]
>
>
```

With our Ethereum node running as a service we can now attach to it using:

```
$ geth attach
```

This gives us an interactive JavaScript console. From here we can call functions, such as:

```
> eth.accounts
```

Which will list the current accounts.

Back, A. (2017). Exploring Ethereum with Raspberry Pi - Part 1: Getting Started.

Start the Node

```
pi@chainpi:~ $
pi@chainpi:~ $ geth attach
Welcome to the Geth JavaScript console!

instance: Geth/v1.7.3-stable-4bb3c89d/linux-arm/go1.7.4
 modules: admin:1.0 debug:1.0 eth:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>
> admin.peers
[{
    caps: ["eth/62", "eth/63", "les/1", "les/2"],
    id: "a979fb575495b8d6db44f750317d0f4622bf4c2aa3365d6af7c284339968eef29b69ad0dce72a4d8db5ebb4968de0e3b
ec910127f134779fbcb0cb6d3331163c",
    name: "Geth/v1.7.3-stable/linux-amd64/go1.7",
    network: {
      localAddress: "10.0.10.100:60604",
      remoteAddress: "52.16.188.185:30303"
    },
    protocols: {
      les: {
        difficulty: 2.2875535680775693e+21,
        head: "70441005f3c27b9c37c7ebeabd75a62c42543740261674e00b14caf30e0017b6",
        version: 2
      }
    }
}]
>
>
```

Or to get information about the connected peers:

```
> admin.peers
```

Note that the light client protocol is still in development, somewhat experimental and does rely on full peers/nodes enabling support for it. As such, it may not be entirely practical at the time of writing to transact on the Ethereum mainnet blockchain using this. That said, things are moving fast and this situation could easily change in the not too distant future.

Back, A. (2017). Exploring Ethereum with Raspberry Pi - Part 1: Getting Started.

OPERATING ETHEREUM ON RASPBERRY PI PROVING ETHEREUM WORKS ON RASPBERRY PI

Stopping Synchronization

Stopping mainnet synchronisation

If you followed along with Part 1 and configured a node to use mainnet and run in light synchronisation mode, this can be stopped and start-up disabled with:

```
$ sudo systemctl stop geth@pi.service

$ sudo systemctl disable geth@pi.service
```


Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Create a New Account

Creating a new account

```
pi@chainpi:~ $
pi@chainpi:~ $ geth --datadir .designspark account new
Your new account is locked with a password. Please give a password. Do not forget thi
s password.
Passphrase:
Repeat passphrase:
Address: {1fd4027fe390abaa49e5afde7896ff1e5ecacabf}
pi@chainpi:~ $
pi@chainpi:~ $
```

We need a name for our new blockchain network and for the purposes of this example, we'll use "DesignSpark". By default Ethereum stores data in a sub-directory of your home directory named ".ethereum", i.e. a hidden directory on Linux/BSD. So as to keep the data for our private blockchain separate, we'll use ".designspark".

If we start by creating a new account:

```
$ geth --datadir .designspark account new
```

And take a note of the address of the account, since we'll need this when we initialise the new network if we would like to preallocate any funds to it.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Initialize The Ethereum Blockchain at Block 0

In the beginning, there was block 0

```
pi@chainpi:~ $
pi@chainpi:~ $ geth --datadir .designspark init designspark.json
INFO [03-03|16:59:17] Allocated cache and file handles         database=/home/pi/.des
ignspark/geth/chaindata cache=16 handles=16
INFO [03-03|16:59:17] Writing custom genesis block
INFO [03-03|16:59:17] Successfully wrote genesis state         database=chaindata
                      hash=acf1f3…047b81
INFO [03-03|16:59:17] Allocated cache and file handles         database=/home/pi/.des
ignspark/geth/lightchaindata cache=16 handles=16
INFO [03-03|16:59:17] Writing custom genesis block
INFO [03-03|16:59:17] Successfully wrote genesis state         database=lightchaindat
a                     hash=acf1f3…047b81
pi@chainpi:~ $
pi@chainpi:~ $
```

There has to be a first link in a chain and a blockchain is no different, requiring a genesis block to be created that will be used by the initial set of nodes which are to participate in the network. This is configured via a JSON file and the contents of the one we used, for example, are below.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

JSON File to Create Block 0

```
{
    "config": {
        "chainId": 555,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
    "difficulty": "20",
    "gasLimit": "2100000",
    "alloc": {
        "1fd4027fe390abaa49e5afde7896ff1e5ecacabf":
        { "balance": "20000000000000000000" }
    }
}
```

The 'chainId' is a numerical value that identifies the network and a list of those currently in use by public networks can be found here. We needed to pick a number for our private DesignSpark network and for some reason 555 seemed like a good choice — you could use a different number.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Other Parameters

- **homesteadBlock.** Homestead is an Ethereum release and for our chain, this is set to 0.
- **eip155Block.** Our chain won't hard-fork for EIP155, so this is set to 0.
- **eip158Block.** Our chain won't hard-fork for EIP158, so this is set to 0.
- **difficulty.** This sets the mining difficulty and in our case, we want this reasonably low.
- **GasLimit.** This is the limit of the Gas cost per block.
- **alloc.** This is where we can pre-allocate funds to accounts.

Ethereum Improvement Proposals (EIPs) describe standards for the Ethereum platform and new ones may be issued to address shortcomings. As a network grows it may be forked at a certain point to allow EIPs to be incorporated. This is not so much a concern for our private network, but for details of where EIP155 was implemented with mainnet and what this does, see Spurious Dragon.

Gas is the unit used as a measure of how much work an action or set of actions takes to perform. Thereby allowing a cost to be attached to executing *smart contracts* — the objects which contain code functions and that live on the blockchain and are able to interact with other contracts, make decisions, store data, and send ether to others. More on this in a future post.

Alloc allows us to preallocate funds to one or more accounts. Here funds have been allocated to the address of the account we created earlier.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Initialize the Network

Having saved our config file to designspark.json we can now initialise the network with:

```
$ geth --datadir .designspark init designspark.json
```

And that's it, we've written out our genesis block and now have the very beginnings of our new network. Provided subsequent nodes are initialised in the same way, they can become members too.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Initialize the Network

Starting the first node

```
INFO [03-03|17:00:10] Disk storage enabled for ethash DAGs      dir=/home/pi/.ethash
               count=2
INFO [03-03|17:00:10] Initialising Ethereum protocol            versions="[63 62]" net
work=555
INFO [03-03|17:00:10] Loaded most recent local header           number=0 hash=acf1f3…0
47b81 td=20
INFO [03-03|17:00:10] Loaded most recent local full block       number=0 hash=acf1f3…0
47b81 td=20
INFO [03-03|17:00:10] Loaded most recent local fast block       number=0 hash=acf1f3…0
47b81 td=20
INFO [03-03|17:00:10] Regenerated local transaction journal     transactions=0 account
s=0
INFO [03-03|17:00:10] Starting P2P networking
INFO [03-03|17:00:10] RLPx listener up                          self="enode://01f5ecc7
c232f7571175bffc71c4e1608e1308e2ce7fd6ed3ae17d5e97e2d5253dcaa854286f99991d671788127f7
902fa56d20875eabae49665a515da105047@[::]:30303?discport=0"
INFO [03-03|17:00:10] IPC endpoint opened: /home/pi/.designspark/geth.ipc
INFO [03-03|17:00:10] HTTP endpoint opened: http://127.0.0.1:8080
Welcome to the Geth JavaScript console!

instance: Geth/chainpi/v1.7.3-stable-4bb3c89d/linux-arm/go1.7.4
coinbase: 0x1fd4027fe390abaa49e5afde7896ff1e5ecacabf
at block: 0 (Thu, 01 Jan 1970 00:00:00 UTC)
 datadir: /home/pi/.designspark
 modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1
.0 web3:1.0

> 
```

To start the first node with the JavaScript console we enter:

```
$ geth --identity chainpi --rpc --rpcport 8080 --rpccorsdomain "*" --da
```


Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Start-up Parameters

What do all the parameters mean?

- **indentity.** This sets the Ethereum node identity.
- **rpc*.** The various RPC settings configure the available APIs and who has access to them.
- **datadir.** We obviously need to use the same data directory as before.
- **nodiscover.** This means our node is not discoverable.
- **networkid.** This needs to be the same numerical ID configured during initialisation.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Node Account Balance

```
> eth.accounts
["0x1fd4027fe390abaa49e5afde7896ff1e5ecacabf"]
>
> primary = eth.accounts[0]
"0x1fd4027fe390abaa49e5afde7896ff1e5ecacabf"
>
> balance = web3.fromWei(eth.getBalance(primary), "ether");
20
>
>
```

Once we've entered the console we can use eth.accounts to list the available accounts and eth.getBalance to check the balance.

```
> eth.accounts

> primary = eth.accounts[0]

> balance = web3.fromWei(eth.getBalance(primary), "ether");
```

Note how the figure returned is much smaller than what we preallocated via designspark.json? That's because the balance in *Ether* was returned, whereas during initialisation the allocation was actually specified in *Wei*, a far smaller unit.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Starting Up Additional Nodes

Creating a 2nd node

```
>
> eth.accounts
["0xbdcb5581d531bf819f48bd74a3f667af240a66a7"]
> primary = eth.accounts[0]
"0xbdcb5581d531bf819f48bd74a3f667af240a66a7"
> balance = web3.fromWei(eth.getBalance(primary), "ether");
0
>
>
```

A blockchain network with only one node wouldn't be much use and so we'll create a second one. This time it's recommended to use a computer with a little more RAM, such as a laptop or desktop running Debian/Ubuntu, as this is likely to be needed should we wish to run a miner at some point.

To recap, the steps involved are:

1. Install geth.
2. Run the command as above to create a new account.
3. Initialise using the same JSON configuration file.
4. Start the node as before, but this time use a different identity!

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Check the Balance on the Newly Added Node

Once we've done this, the node has been started and dropped into the JavaScript console, we can then once again check the new account and its balance with:

```
> eth.accounts

> primary = eth.accounts[0]

> balance = web3.fromWei(eth.getBalance(primary), "ether");
```

This time we should see we have a balance of 0, as we didn't preallocate any funds to the account.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Starting Up the Node

To start the node:

$ geth --identity raspberrypi1 --rpc --rpcport 8080 --rpccorsdomain "*" --datadir .designspark --port 30302 --nodiscover --rpcapi "db,eth,net,web3" --networkid 555 console

To check the balance that we allocated:

>eth.accounts

```
pi@raspberrypi:~ $ geth --identity raspberrypi1 --rpc --rpcport 8080 --rpccorsdomain "*" --datadir .designspark --port 30302 --nodiscover --rpcapi "db,eth,net,web3" --networkid 555 console
INFO [04-19|18:55:44] Starting peer-to-peer node               instance=Geth/raspberrypi1/v1.7.3-stable-4bb3c89d/linux-arm/go1.7.4
INFO [04-19|18:55:44] Allocated cache and file handles         database=/home/pi/.designspark/geth/chaindata cache=128 handles=1024
WARN [04-19|18:55:44] Upgrading database to use lookup entries
INFO [04-19|18:55:44] Database deduplication successful        deduped=0
INFO [04-19|18:55:44] Initialised chain configuration          config="{ChainID: <nil> Homestead: 0 DAO: <nil> DAOSupport: false EIP150: <nil> EIP155: 0 EIP158: 0 Byzantium: <nil> Engine: u
nknown}"
INFO [04-19|18:55:44] Disk storage enabled for ethash caches   dir=/home/pi/.designspark/geth/ethash count=3
INFO [04-19|18:55:44] Disk storage enabled for ethash DAGs     dir=/home/pi/.ethash                count=2
INFO [04-19|18:55:44] Initialising Ethereum protocol           versions="[63 62]" network=555
INFO [04-19|18:55:44] Loaded most recent local header          number=0 hash=42041b…fec43d td=20
INFO [04-19|18:55:44] Loaded most recent local full block      number=0 hash=42041b…fec43d td=20
INFO [04-19|18:55:44] Loaded most recent local fast block      number=0 hash=42041b…fec43d td=20
INFO [04-19|18:55:44] Regenerated local transaction journal    transactions=0 accounts=0
INFO [04-19|18:55:44] Starting P2P networking
INFO [04-19|18:55:44] RLPx listener up                         self="enode://07170f08de0e608cf8c0eeaaf5466518264cdaa685034b696f12949011b1398b1bc2f74aef39bb1681d722f8b96b7dbb2951da62201ea860e
65c4e099b069791f@[::]:30302?discport=0"
INFO [04-19|18:55:44] IPC endpoint opened: /home/pi/.designspark/geth.ipc
INFO [04-19|18:55:44] HTTP endpoint opened: http://127.0.0.1:8080
Welcome to the Geth JavaScript console!

instance: Geth/raspberrypi1/v1.7.3-stable-4bb3c89d/linux-arm/go1.7.4
coinbase: 0x9fc1843c34bcc15e926a2f300748aaaec44a406c
at block: 0 (Thu, 01 Jan 1970 00:00:00 UTC)
 datadir: /home/pi/.designspark
 modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> INFO [04-19|18:55:46] Mapped network port                    proto=tcp extport=30302 intport=30302 interface="UPNP IGDv2-IP1"
> eth.accounts
["0x9fc1843c34bcc15e926a2f300748aaaec44a406c"]
> admin.peers
[]
> eth.getBalance("9fc1843c34bcc15e926a2f300748aaaec44a406c")
2000000000000000000
```


Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Creating the Second Node

Creating 2nd node:

- Follow the same steps as mentioned above for node 1.
- Create an account
- Use the same .json file.

Connect the Peers

Connecting the peers

```
>
> admin.nodeInfo.enode
"enode://5156218119a3697389a34bf0a19ceca49d9f3d06948836b8cc6c206c9f7b7081e64537eeb0f9
c059561736a8e7cb6ebbe438028dd949d0f69f4cab642c11d46c@[::]:30303?discport=0"
>
>
```

Since we don't want our nodes to be discoverable we started them with the --nodiscover option, meaning that we'll need some way of configuring them to peer. We can achieve this by creating a file called static-nodes.json located in the datadir, which in our case is ~/.designspark.

First, though we need to get the *enode URL* for each of our nodes by entering at the JavaScript console on each system:

```
> admin.nodeInfo.enode
```

We then populate the static-nodes.json file with this info as follows:

```
[
"enode://01f5ecc7c232f7571175bffc71c4e1608e1308e2ce7fd6ed3ae17d5e97e2d
]
```


Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Connecting Peers

Note how [::] has been replaced by the node IP address and the ?discport=0 suffix omitted.

```
>
> admin.peers
[{
    caps: ["eth/63"],
    id: "5156218119a3697389a34bf0a19ceca49d9f3d06948836b8cc6c206c9f7b7081e64537eeb0f9
c059561736a8e7cb6ebbe438028dd949d0f69f4cab642c11d46c",
    name: "Geth/snow/v1.8.0-stable-5f540757/linux-amd64/go1.9.4",
    network: {
      localAddress: "10.100.1.196:30303",
      remoteAddress: "10.100.1.229:41152"
    },
    protocols: {
      eth: {
        difficulty: 20,
        head: "0xacf1f3c3898431e37b0c07c7421c203d9a90475a51b8d1f2c7048de207047b81",
        version: 63
      }
    }
}]
>
```

Once this file has been created on both nodes we can exit geth via CTRL-D and then re-launch the console. Following which if we enter on the first node:

```
> admin.peers
```

We should see the details for the second node.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Peer Verification

```
> admin.peers
```

We should see the details for the second node.

```
>
> admin.peers
[{
    caps: ["eth/63"],
    id: "01f5ecc7c232f7571175bffc71c4e1608e1308e2ce7fd6ed3ae17d5e97e2d5253dcaa854286f
99991d671788127f7902fa56d20875eabae49665a515da105047",
    name: "Geth/chainpi/v1.7.3-stable-4bb3c89d/linux-arm/go1.7.4",
    network: {
      inbound: false,
      localAddress: "10.100.1.229:41152",
      remoteAddress: "10.100.1.196:30303",
      static: true,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 20,
        head: "0xacf1f3c3898431e37b0c07c7421c203d9a90475a51b8d1f2c7048de207047b81",
        version: 63
      }
    }
}]
>
```

Repeating this on the second node we should then see the node info for the first.

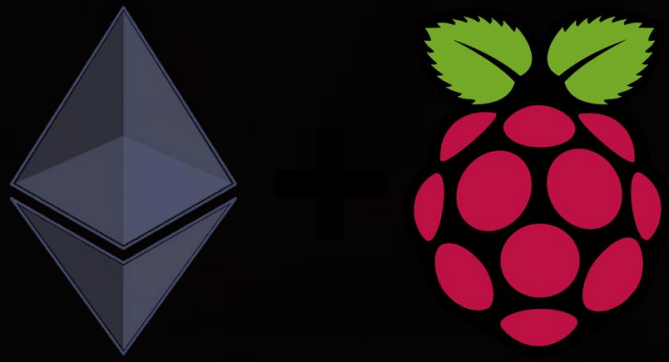
So now we have our own private blockchain network complete with two nodes, each configured with an account and one of those with preallocated funds.

Back, A. (2017). Exploring Ethereum with Raspberry Pi Part 2: Creating a Private Blockchain.

Conclusion

- **Blockchain:**
  - A technical marvel made possible by software, hardware, strong cryptography, and the Internet
  - Has made significant progress in only 100+ months
  - Has significant strengths and a few limitations too
  - Blockchain is starting to be widely used to automate trusted computing transactions and increase efficiencies in many industries
  - Has great potential because of popular support of talented nerds, and now major players in major industries

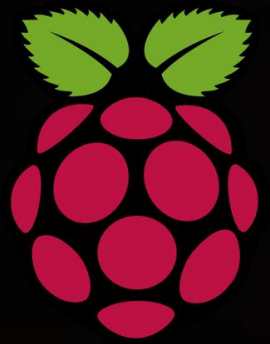
- The excitement about the blockchain is based on its ability to serve as a tool for achieving and maintaining integrity in purely distributed peer-to-peer systems that have the potential to change whole industries due to disintermediation.



Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Conclusion

- **Ethereum and Raspberry :**
  - Ethereum is FREE
  - Raspberry Pi devices are cheap, plentiful, and easy to purchase at places like www.amazon.com .
  - There is a wealth of free information available on how to use these important technologies.
  - Everyone who is interested in Blockchain and Ethereum should consider learning Blockchain and Raspberry Pi, so they will be familiar with and be able to use it as a spring board to learn Blockchain and Blockchain Development.
- **Chicago Blockchain Community:**
  - Join us, participate and get involved.
  - Chicago Blockchain Project
    - **https://www.meetup.com/chicagoblockchainproject/**
  - Chicago Bitcoin and Open Blockchain Meetup
    - **https://www.meetup.com/Bitcoin-Open-Blockchain-Community-Chicago/**



Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Resources

Cameroon:
“Africa in Miniature”

Resources - Cameroon

- Akpan, M. (2016) Cameroon Art and Culture: Ethnic groups, Religion, Tradition, Tribes, History and People. Published by CreateSpace Independent Publishing Platform.
- CIA. (2019) Cameroon World Fact Book: Retrieved from https://www.cia.gov/library/publications/the-world-factbook/geos/cm.html on July 8, 2020.
- COVID-19 Statistics. (2020) COVID-19 statistics for Cameroon. Retrieved from https://www.covid19.onl/country/cameroon on July 14, 2020.
- Displore. (2020). An HONEST Explanation of the ANGLOPHONE CRISIS in Cameroon. Retrieved from https://www.youtube.com/watch?v=JQXK1B_0IaM on July 10, 2020.
- Etoundi, R. A., et al. (2016). Development of the Digital Economy in Cameroon: Challlenges and Perspectives. Retrieved from https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1681-4835.2016.tb00558.x on July 9, 2020.

Resources - Cameroon

- Hamel, P. J. and Guien, D. (2017). Cameroon Travel Adventure. Published by CreateSpace Independent Publishing Platform.
- Operation World. (2019). Cameroon. Published by IVPress.com. Retrieved from http://www.operationworld.org/print/141 on July 8, 2020.
- Sosale, S. and Majgaard, K. (2016). Fostering Skills in Cameroon: Inclusive Workforce Development, Competitiveness, and Growth (Directions in Development) 1st Ed. World Bank Publications.
- Tchouteu, J., et al. (2017). CAMEROON: The Haunted Heart of Africa. Published independently.
- West, B. (2011). Cameroon. Published by Bradt Travel Guides.

Additional Resources

Additional Resources

- Antonopoulos, A. M. (2018). Mastering Bitcoin: Programming the Open Blockchain, second edition. Sebastopol, CA: O'Reilly Media, Inc.
- Antonopoulos, A. M. and Wood, G. (2019). Mastering Ethereum: Building Smart Contract sand DApps. Sebastopol, CA: O'Reilly Media, Inc.
- Axios. (2020). May Meeker's COVID-19 Trends Report. Retrieved from https://www.axios.com/mary-meeker-coronavirus-trends-report-0690fc96-294f-47e6-9c57-573f829a6d7c.html on June 27, 2020.
- Bahga, A. and Madisetti, V. (2017). Blockchain Applications: A Hands-On Approach. Published by Arshdeep Bahga and Vijay Madisetti. www.blockchain-book.com .
- Bambara, J. J. and Allen P. R. (2018). Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions. New York, NY: McGraw-Hill Education.
- Bashir, I. (2018). Mastering Blockchain, second edition. Birmingham, UK: Packt Publishing Ltd.
- **Bharadwaj, H. (2018). New Realities: VR, AR, MR, and the Future of Design. Retrieved from https://www.toptal.com/designers/product-design/vr-ar-mr-the-future-of-design on July 17, 2020.**
- Blockchain Training Alliance. (2019). Global Blockchain Terms, version 2.0. Retrieved from https://cdn.shopify.com/s/files/1/2137/1081/files/BTA_Global_Blockchain_Terms.pdf?2499 on August 14, 2019
- Casey, M. J. and Vigna, P. (2018). The Truth Machine: The Blockchain Reference and the Future of Everything. New York, NY: St. Martin's Press.
- Caughey, M. (2013). Bitcoin Step by Step, second edition. Amazon Digital Services.

Additional Resources

- CB Insights. (2018) Memes That Kill: The Future of Information Warfare. Retrieved on May 10, 2018 from https://app.cbinsights.com/research/future-of-information-warfare/ .
- Champagne, P. (2014). The Book of Satoshi: The Collected Writings of Bitcoin Creator Satoshi Nakamoto. Published by E53 Publishing, LLC.
- Chopra-McGowan, A. and Reddy, S. B. (2020). What Would It Take to Reskill Entire Industries? Retrieved from https://hbr.org/2020/07/what-would-it-take-to-reskill-entire-industries on July 14, 2020.
- Dannen, C. (2017). Introducing Ethereum and Solidity: Foundations of Crytocurrency and Blockchain Programming for Beginners. New York, NY: Apress
- De Filippi, P. and Wright, A. (2018). Blockchain and the Law: the Rule of Code. Cambridge, MA: President and Fellows of Harvard College.
- De Havilland, P. (2018). Greedy, Prodigal, and Suicidal — Hosho to Save Smart Contracts From Three Deadly Sins. An article published at Bitsonline.com on September 3, 2018. Retrieved from https://bitsonline.com/greedy-prodigal-suicidal-hosho-smart-contracts/ on February 27, 2019.
- Dhillon, V., Metcalf, D., and Hooper, M. (2017). Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Nake It Work for You. New York, NY: Apress.
- Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.
- Eddison, L. (2017). Ethereum: A Deep Dive into Ethereum. Published by Leonard Eddison.
- Etwaru, R. (2017). Blockchain Trust Companies. Indianapolis, IN: Dog Ear Publishing.
- Ferry, T. (2019). To Blockchain or not to Blockchain. An article published at Medium.com on June 8, 2018. Retrieved on January 13, 2019 from https://medium.com/causys/to-blockchain-or-not-to-blockchain-aed05bf08150 .

Additional Resources

- Infante, R. (2019) Building Ethereum DApps. Shelter Island, NY: Manning Publications.
- Langreth, R and Court, E., (2020). Brain Deficits, Nerve Pain Can Torment COVID Patients for Months. Retrieved from https://www.bloomberg.com/news/articles/2020-08-26/brain-deficits-nerve-pain-as-covid-torments-infected-for-months?sref=Twek4M08 on August 26, 2020.
- Laurence, T. (2017). Blockchain for Dummies. Hoboken, NJ: John Wiley & Sons, Inc.
- Lee, T. B. (2013). 12 questions about Bitcoin you were too embarrassed to ask. Retrieved from http://www.washingtonpost.com/blogs/the-switch/wp/2013/11/19/12-questions-you-were-too-embarrassed-to-ask-about-bitcoin/ on November 19, 2013.
- Lombana-Bermudez, A., et al. (2020) "Youth and the Digital Economy: Exploring Youth Practices, Motivations, Skills, Pathways, and Value Creation," Youth and Media, Berkman Klein Center for Internet & Society Published by the Harvard Library for Scholarly Communication. Retrieved from https://dash.harvard.edu/handle/1/42669835 on July 11, 2020.
- Ma, M. (2017). Blockchain Design Sprint: An Agile Innovation Workbook to Implement an Agile Design Sprint for your Blockchain Business. Published by Future Lab www.futurelabconsulting.com.
- MANRS. (2017). Mutually Agreed Norms for Routing Security (MANRS) Implementation Guide, version 1.0 Retrieved on April 14, 2020 from https://www.manrs.org/isps/guide/.
- Moore, G. E. (1965). Cramming More Components onto Integrated Circuits. Published in Electronics Magazine, April 1965. Retrieved from http://www.billslater.com/moore's_law_1965.pdf on July 17, 2020.
- Nakamoto. S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf on November 1, 2013.

Additional Resources

- Navarro, M. (2020). What Would It Take to Reskill Entire Industries? Retrieved from https://blog.iammarketingmedia.com/what-would-it-take-to-reskill-entire-industries/ on July 14, 2020.
- Nguyen, J. (2019). Blockchain still vulnerable to hacks despite security hype, but here are some solutions. Retrieved from https://e27.co/blockchain-still-vulnerable-to-hacks-despite-security-hype-but-here-are-some-solutions-20190212/ on February 13, 2019.
- O'Ham, T. (2018). Singapore Research Team Codifies 3 new Ethereum VM Vulnerabilities. An article published at Bitsonline.com on February 21, 2018. Retrieved from https://bitsonline.com/singapore-research-ethereum/ on February 27, 2019.
- Orcutt, M. (2019). Once Hailed as Unhackable, Blockchains Are now Getting Hacked. An article in MIT Review. Published February 19, 2019. Retrieved from https://www.technologyreview.com/s/612974/once-hailed-as-unhackable-blockchains-are-now-getting-hacked/ on February 24, 2019.
- Popper, N. (2013). Into the Bitcoin Mines, Retrieved from http://dealbook.nytimes.com/2013/12/21/into-the-bitcoin-mines/?hp&_r=0 on December 21, 2013.
- Prusty, N. (2017). Building Blockchain Projects: Building Decentralized Blockchain Applications with Ethereum and Solidity. Birmingham, UK: Pact Publishing.

Additional Resources

- Ramone, A. D. (2019). How to Secure a Blockchain: 3 Things Business Leaders Know. An article published at Techrepublic.com on April 18, 2019. Retrieved from https://www.techrepublic.com/article/how-to-secure-a-blockchain-3-things-business-leaders-need-to-know/ on April 23, 2019.
- Randall, I. (2020). Global internet outages reach a record high during the coronavirus lockdown as broadband operators tinker with networks to meet increased demand from people working from home. Published April 29, 2020 at Daily Mail UK. Retrieved from https://www.dailymail.co.uk/sciencetech/article-8269245/Global-internet-outages-reach-record-high-coronavirus-lockdown.html on April 30, 2020.
- SCGNEWS. (2014). The IRS Just Declared War on Bitcoin - Retroactively. Retrieved from http://scgnews.com/the-irs-just-declared-war-on-bitcoin-retroactively on March 27, 2014.
- Schudel, G. and Smith, D.J. (2008). Router Security Strategies: Securing IP Traffic Planes. Indianapolis, IN: Cisco Press.
- Slater, W. F. (2002). The Internet Outage and Attacks of October 2002. Retrieved from http://www.billslater.com/writing/2002_1107__Internet_Outage_and_Attacks_in_october_2002_by_William_Slater.pdf on May 1, 2020.

Additional Resources

- Slater, W. F. (2020). A Proposal to Improve MANRS Global Secure Internet Routing Policy Management Using Blockchain Technology. Retrieved from http://www.billslater.com/manrs/ISOC_MANRS_Chapter%20_Initiative_Project_William_Slater_2020_0606_v01.1_.pdf on June 26, 2020.
- Slater, W. F. (2020). Leadership in Digital Uncertainty: New Strategy to Cope with the New Normal. Retrieved from http://www.billslater.com/cameroon_2020.pdf on July 17, 2020.
- Slater, W. F. (2020). Digital Transformation Topics, Perspectives and Opportunities to Help Cameroon Succeed and Thrive in the Post COVID-19 World - Prepared for the CAPDA Conference in Yaounde & Libreville, Cameroon (West Africa), July 23, 2020. Retrieved from http://www.billslater.com/cameroon_202007.pdf .
- Slater, W. F. (2020). A Proposal to Improve MANRS Global Secure Internet Routing Policy Management Using Blockchain Technology. Write as the the Chapter Initiative Projecct Requirement for the ISOC Course on Mutually Assured Norms for Routing Security (MANRS). Retrieved from http://www.billslater.com/manrs/ISOC_MANRS_Chapter%20_Initiative_Project_William_Slater_2020_0606_v01.1_.pdf on June 26, 2020.

Additional Resources

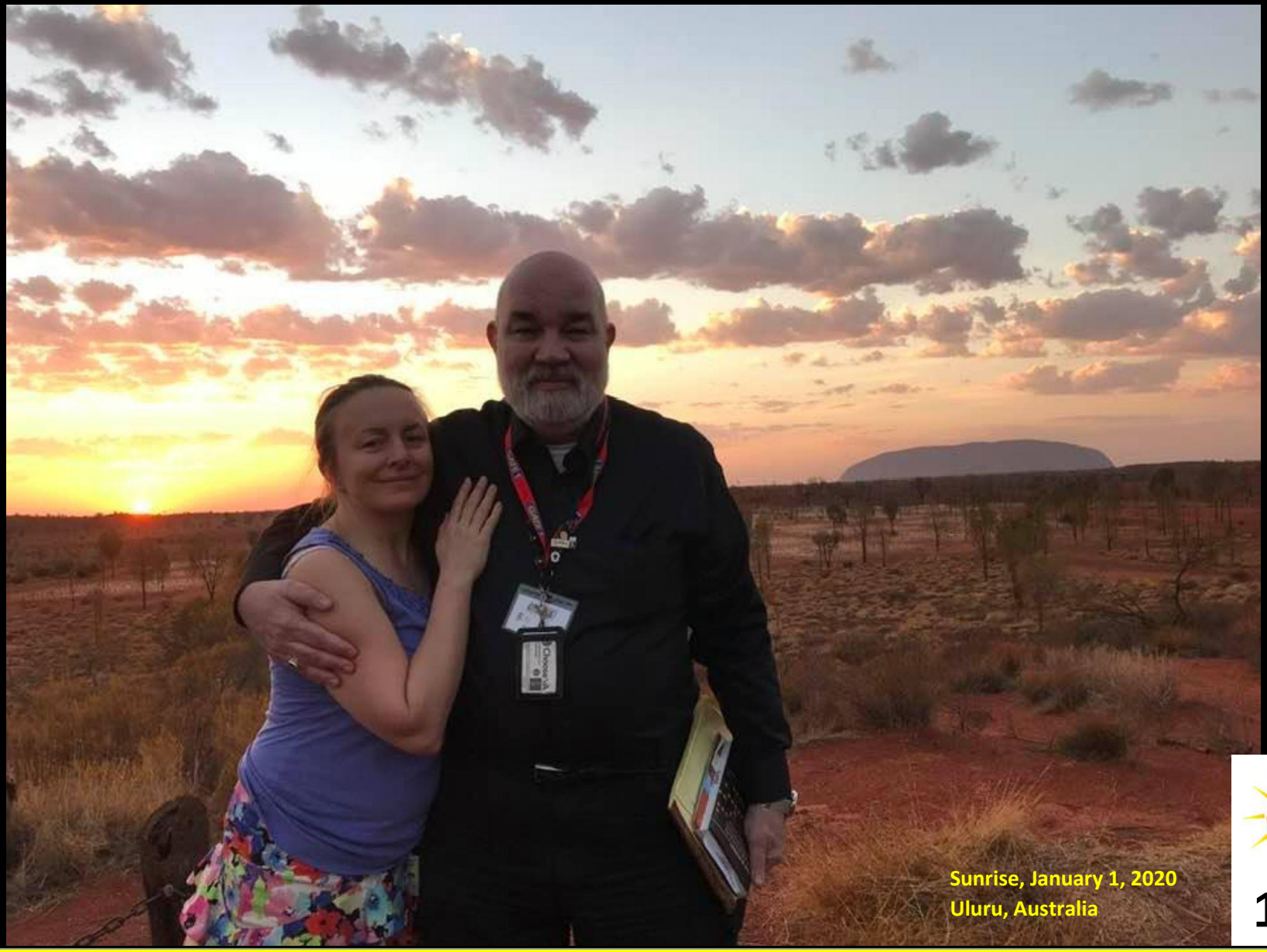
- Slater, W. F. (2020). A Proposal to Improve MANRS Global Secure Internet Routing Policy Management Using Blockchain Technology. Retrieved from http://www.billslater.com/manrs/ISOC_MANRS_Chapter%20_Initiative_Project_William_Slater_2020_0606_v01.1_.pdf on June 26, 2020.
- Slater, W. F. (2019). How to "Blockchain" Your IT Organization. Retrieved from http://www.authorstream.com/Presentation/billslater-3536040-roadmap-blockchain-organization-square/ on July 17, 2020.
- Slater, W. F. (2015). Telework: Risks, Challenges, Perils, and Successes. Retrieved from http://www.billslater.com/writing/teleworking2015_slater.pdf on July 17, 2020.
- Slater, W. F. (2002). The Internet Outage and Attacks of October 2002. Retrieved from http://www.billslater.com/writing/2002_1107__Internet_Outage_and_Attacks_in_october_2002_by_William_Slater.pdf on May 1, 2020.
- Smith, T. (2020) A Supercomputer Analyzed Covid-19 — and an Interesting New Theory Has Emerged. A closer look at the Bradykinin hypothesis. Retrieved from https://elemental.medium.com/a-supercomputer-analyzed-covid-19-and-an-interesting-new-theory-has-emerged-31cb8eba9d63 on September 1, 2020.

Additional Resources

- Smith, B. (2019). The Evolution of Cryptocurrency in Terrorism. Retrieved from Blockchain Training Alliance. (2019). Global Blockchain Terms, version 2.0. Retrieved on August 14, 2019 from https://www.bellingcat.com/news/2019/08/09/the-evolution-of-bitcoin-in-terrorist-financing/ on August 10, 2019.
- Xu, X., Weber, I, and Stables, M. (2019). Architecture for Blockchain Applications. Nature, Switzerland: Springer Publications.
- Zenko, M. (2017). Bitcoins for Bombs – a Blog published at the Council on Foreign Relations on August 17, 2017. Retrieved from https://www.cfr.org/blog/bitcoin-bombs on February 13, 2019.

Dedication

Dedicated with Love and Everlasting Devotion to My Lovely Bride, Joanna Roguska, Who Is the Incarnate Holy Angel Who Our Lord God Miraculously Placed in My Life in 2000 to Rescue, Love, Inspire and Guard Me on a Daily Basis from then to Eternity.



Sunrise, January 1, 2020
Uluru, Australia