Ethereum for Resource Constrained Devices

Bob Summerwill

DEVCON2, Shanghai

19th Sep 2016

Introducing Bob Summerwill

- 20+ years as a professional developer
- 18+ years in the games industry
- 15+ years at EA, mainly EA Sports
- Config management, architecture
- Software engineering generalist
- CEO of doublethinkco
    - Working on ARM Linux C++ cross-builds
    - Got Wanxiang Blockchain Labs grant


bob-pi3
Thu 15 Sep, 18:17
en_CA
bob
Log In
ubuntu MATE
LG
ODROID


5:41

What are resource-constrained devices?

- Limited CPU (32-bit, single core, slow)
- Limited memory (256MB or 512MB)
- Limited storage (4GB flash or less)
- Limited network connectivity (BLE, 6LowPAN)
- Limited power (running on batteries)
- Limited operating system (single application)

Obviously all devices are “constrained”, but this term usually refers to constraints in comparison to a general purpose personal computer. It can also refer to fixed-specification hardware which cannot be upgraded

Examples

- Wearables (smart rings, smart watches)
- Mobile
- Tablets
- Games consoles
- Embedded devices
- Single board computers (SBC), like Raspberry Pi
- IoT devices of all stripes

Why would you want Ethereum in such devices?

- Mobile computing **is** now mainstream computing
- Wearable computing is the next phase of that miniaturization
- Edge computing required to reach multi-billion-node scale
- Autonomous agents for emergent behaviour
- “Sewer Rat” not “Bubble Boy” security - thanks to Andreas for the analogy!

1998 - Java Ring - Dallas Semiconductors

1998 - Java Ring - Apple Pay + smart locks

- There is nothing new in our desire for security and privacy
- Java Ring was released in 1998
- Running JavaCard 2.0 (embedded Java variant)
- 1024-bit RSA encryption
- 128K RAM
- Digital Signature Standard (FIPS 186)
- 10-20 MHz processor
- 10 year battery life

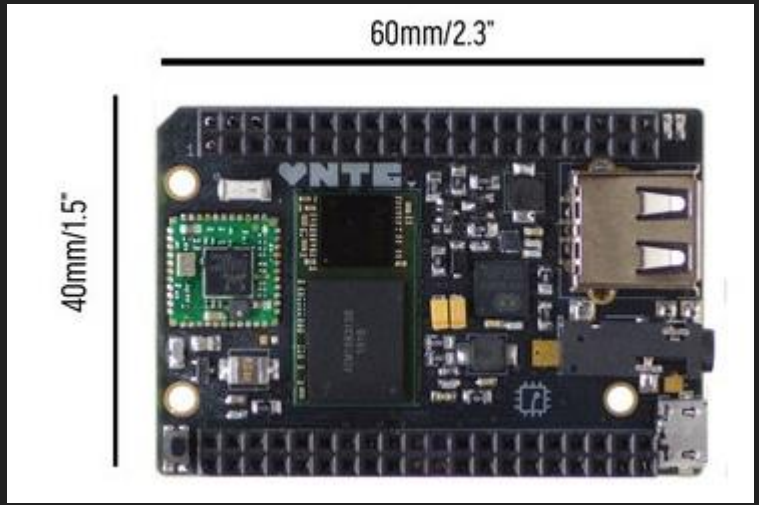
1998 - Blue Dot

2016 - Samsung Gear S3 Frontier smartwatch (~$300 USD)

- Exynos 3250 Dual
  (Dual-core 1.0 GHz Cortex-A7)
- 768MB of memory
- 4GB of flash storage
- LTE, WiFi and BLE connectivity, GPS
- WPC wireless charging, IP68 water resis.
- Accelerometer, gyro, heart rate, barometer
- Speaker and microphone

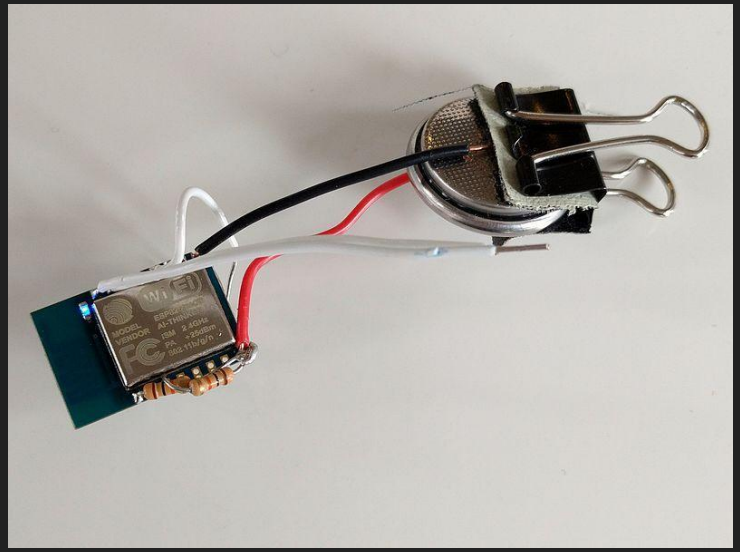
2016 - Project C.H.I.P ($9 USD)

- SoC – Allwinner R8 Cortex A8 processor @ 1 GHz with Mali-400 GPU (Compatible with Allwinner A13)
- System Memory – 512 MB RAM Storage
- 4GB NAND flash
- Connectivity – 802.11 b/g/n Wi-Fi + Bluetooth 4.0
- Video Output – 3.5mm jack for composite video and audio (HDMI and VGA available via adapters)
- USB – 1x USB host port, 1x micro USB OTG port



2016 - ESP8266 - Espressif Systems (~$2 USD)

- 32-bit RISC CPU:
- Tensilica Xtensa LX106 running at 80 MHz
- 64 KiB of instruction RAM
- 96 KiB of data RAM
- External QSPI flash - 512 KiB to 4 MiB*
    - (up to 16 MiB is supported)



50 Billion IoT Devices by 2020?

Since they first made their projections, both Ericsson and Evans have lowered their expectations from 50 billion for 2020: Evans, who is now CTO of Stringify, says he expects to see **30 billion** connected devices by then, while Ericsson figures on **28 billion by 2021**. Other firms have adopted similar tones: IHS Markit projects **30.7 billion** IoT devices for 2020, and Gartner expects **20.8 billion** by that time (excluding smartphones, tablets, and computers). Lastly, IDC anticipates **28.1 billion** (again, not counting those devices).

Meanwhile, the popular 50 billion figure continues to be widely cited. Even Evans is a bit surprised by its lasting power. “I think people do tend to latch onto numbers that seem really hard to fathom,” he says. **“Fifty billion is pretty staggering.”**

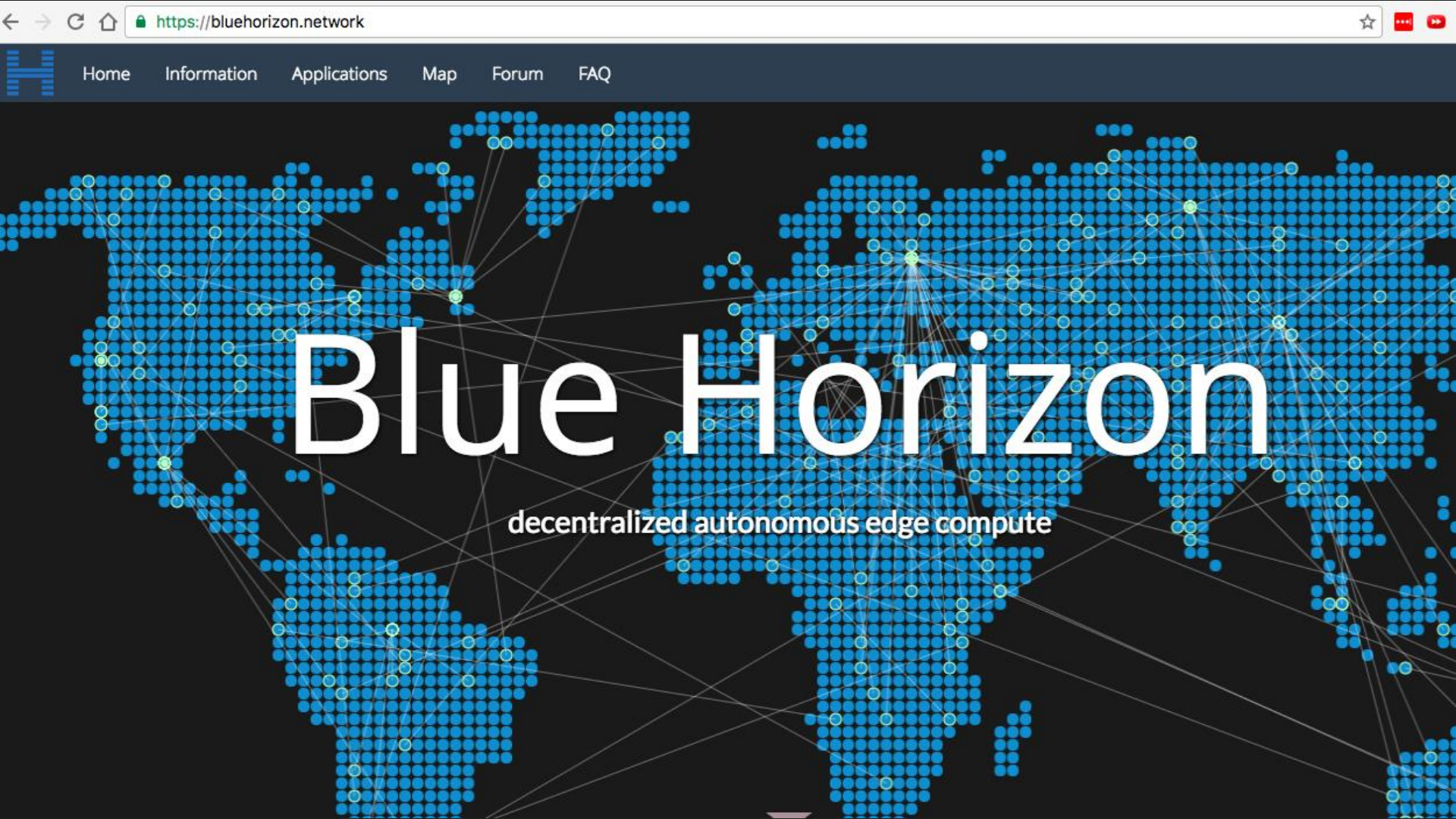
Sounds familiar?

- Are you thinking ... “Have I seen this presentation before?”


BL■CKCHAIN
W■RKSH■PS

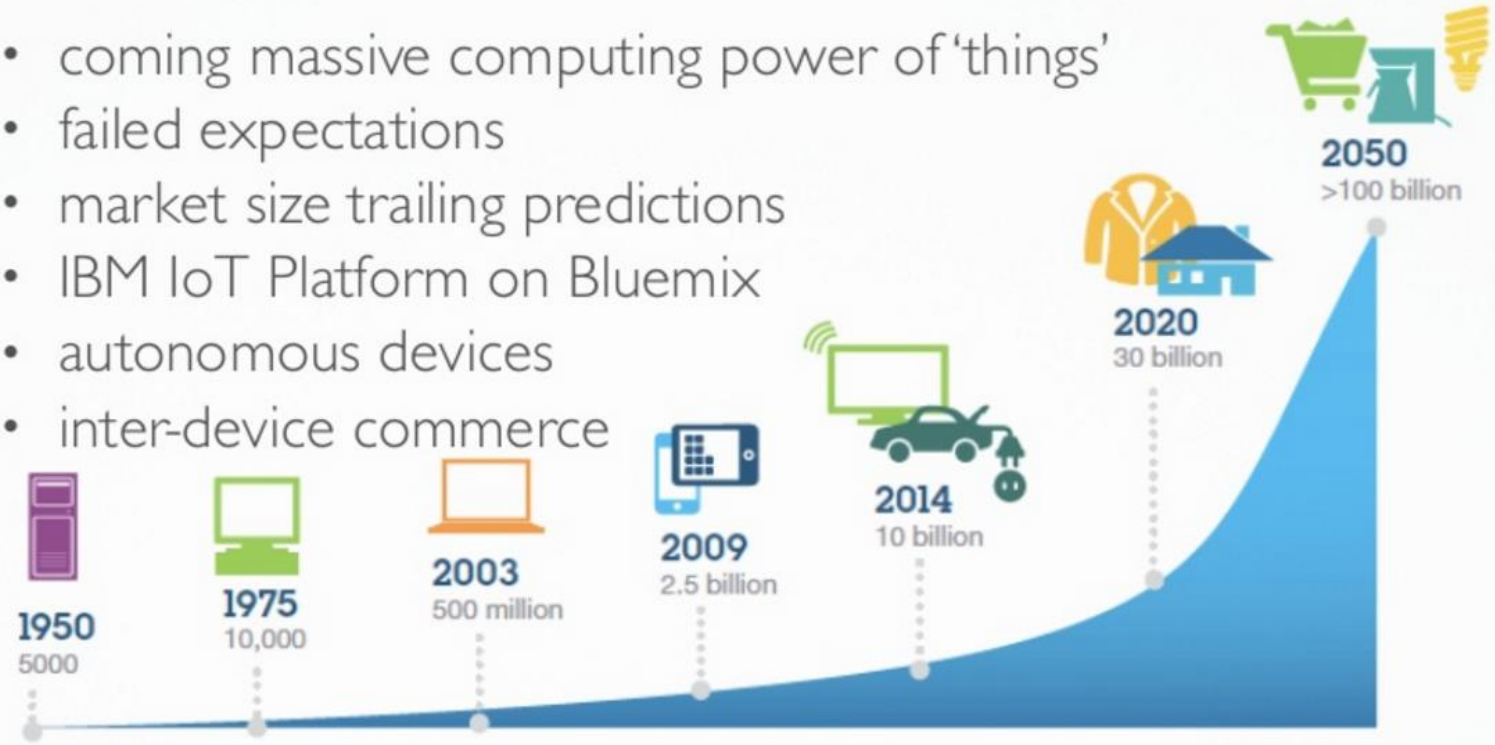
Sounds familiar?

- IBM/Samsung Project ADEPT
  - Unveiled at CES in Las Vegas in January 2015
  - Demonstration on TheProtocol.TV in February 2015
  - Using TeleHash, BitTorrent and Ethereum
  - The launch of IBM’s initiative for IoT
- IBM MTN Project
  - Presentation by Henning Diedrich at DEVCON1 in November 2015
- IBM Blue Horizon Project
  - Ongoing, Henning Diedrich


https://bluehorizon.network
Home
Information
Applications
Map
Forum
FAQ
Blue Horizon
decentralized autonomous edge compute

BIG PICTURE IOT

- coming massive computing power of 'things'
- failed expectations
- market size trailing predictions
- IBM IoT Platform on Bluemix
- autonomous devices
- inter-device commerce



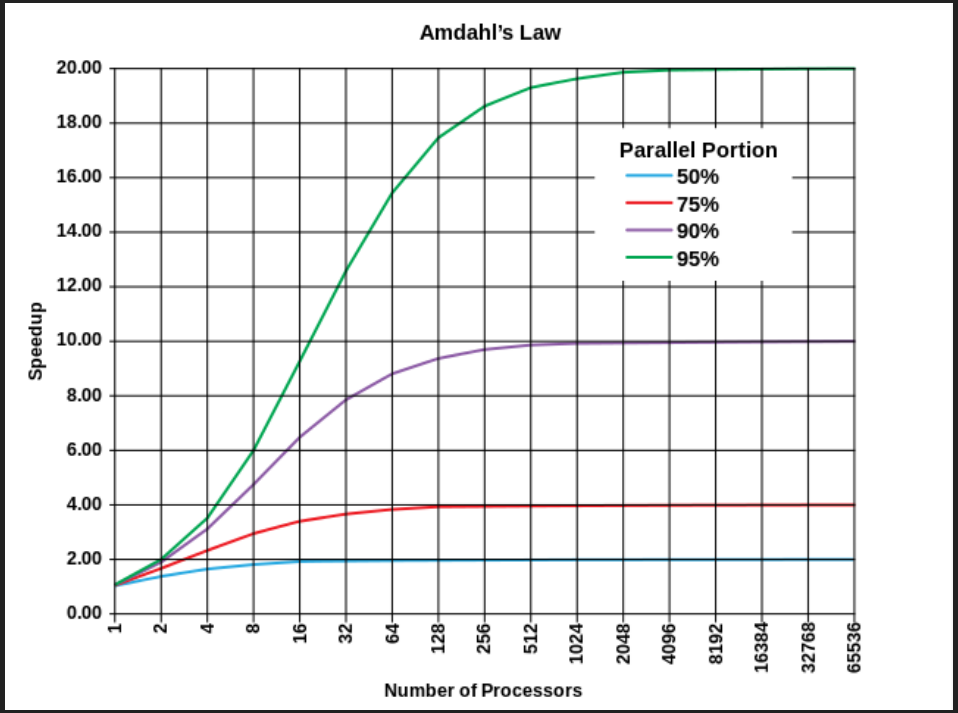
IBM

Ethereum - Sliding scale of options

- Sign transactions offline and talk to “trusted server”
- Light client (LES)
- Full node
- Archival node

Why C++ for resource-constrained devices?

- Moore’s Law is slowing down
  - 7nm is end of the line?
- Amdahl’s Law
- Raw performance does matter
- Optimization for power
- Modern C++ does not suck
- Maximal portability



cross-cpp-ethereum

- Dockerfiles and bash scripts to cross-build cpp-ethereum for ARM Linux
- Development started in July 2015
- First successful cross-build binaries in November 2015
- Largely stalled since February, pending cpp-ethereum reboot and code reorg
- Development efforts have recently re-started
- The go-ethereum LES implementation is nearing integration
- The big TODO is for a C++ implementation of LES sub-protocol
- BlockGrantX Round #2 funding winner

Devices with known cross-cpp-ethereum success

- Mobile
  - Jolla Phone (Sailfish OS)
  - Meizu MX4 Ubuntu Edition (Ubuntu Phone)
- SBCs
  - Raspberry Pi (Model A, B, B+, 2, 3, Zero)
  - Beaglebone Black
  - Odroid XU3
  - Project C.H.I.P.
  - Wandboard Quad

cross-cpp-ethereum - work in progress

- More SBCs
- Tizen devices (Gear smartwatches, Samsung Z phones, Artik SBCs)
- Android
- iOS and tvOS and watchOS
- x86 devices (very low demand)

go-ethereum cross builds

- Being generated nightly, using xgo
- Courtesy of Péter Szilágyi (karalabe)
- Multiple architectures:
  - ARMv7
  - ARM64
  - MIPS64

go-ethereum Light Client (LES)

- Developed by Zsolt Felföldi
- Presentation from him **tomorrow**
- LES on Raspberry Pi 3
  - geth binary: 25MB
  - chain data: 54.9MB
  - cpu: 12%
  - ram: 162MB
  - header sync: 9min

And we’re still waiting on Whisper too!

- Development has recently restarted
- Vlad Glukhovsky (who did much of the work on the earlier C++ version) is now working on a new major revision in go-ethereum
- https://gitter.im/ethereum/whisper
- Coming soon!

Ultimate constraints

- How small can we go?
  - Raspberry Pi Model A smallest so far
  - ARMv6 700Mhz, single core, 256MB of memory
- CPU has to be able to execute the EVM fast enough
- You need to have enough memory to process the per-contract storage
- You need to have enough storage to store blocks (and performance of storage can really matter)
- Network connection needs to be able to keep up with the chain

Conclusion

- We are going to see Ethereum on a huge number of devices
- Many of those devices will be significantly less powerful than desktops
- Ethereum is going to get an awful lot faster and less resource-hungry
- IoT and blockchain is a dream
- IoT without blockchain is a security nightmare

Next steps?

ARM binaries:

- http://doublethink.co
  - https://github.com/doublethinkco/cpp-ethereum-cross
- http://ethembedded.com
  - https://github.com/ethembedded

Light Client (LES)

- https://gitter.im/ethereum/light-client

Thanks for your time!