RuyiSDK: Get ready for 1 million RISC-V software developers

Wei Wu (吴伟)
wuwei2016@iscas.ac.cn
Director of PLCT Lab, ISCAS

About Me (@lazyparser)

- RISC-V ambassador (^o^)/
- **PLCT Lab** (2019-): Compilers, Runtimes, and Simulators
  - TARSIER Project (2022-): Make RISC-V a tier-1 support for FOSS community.
- Member of RISC-V International Foundation (RVI) Technical Committee (TSC)
- RISC-V China Community (CNRV) Core Organizer (2020-)
- Member of LLVM BoD (2022-)
- Chair of HelloGCC(2013-) and HelloLLVM (2018-2022)

Agenda

1. RISC-V is the future. We're seeing it.
2. Why RuyiSDK: The Motivation
3. The Mission of RuyiSDK
4. RuyiSDK architecture and components
5. Beyond RuyiSDK

RISC-V is the future. We're seeing it.

Three Basic Observations (Axioms)

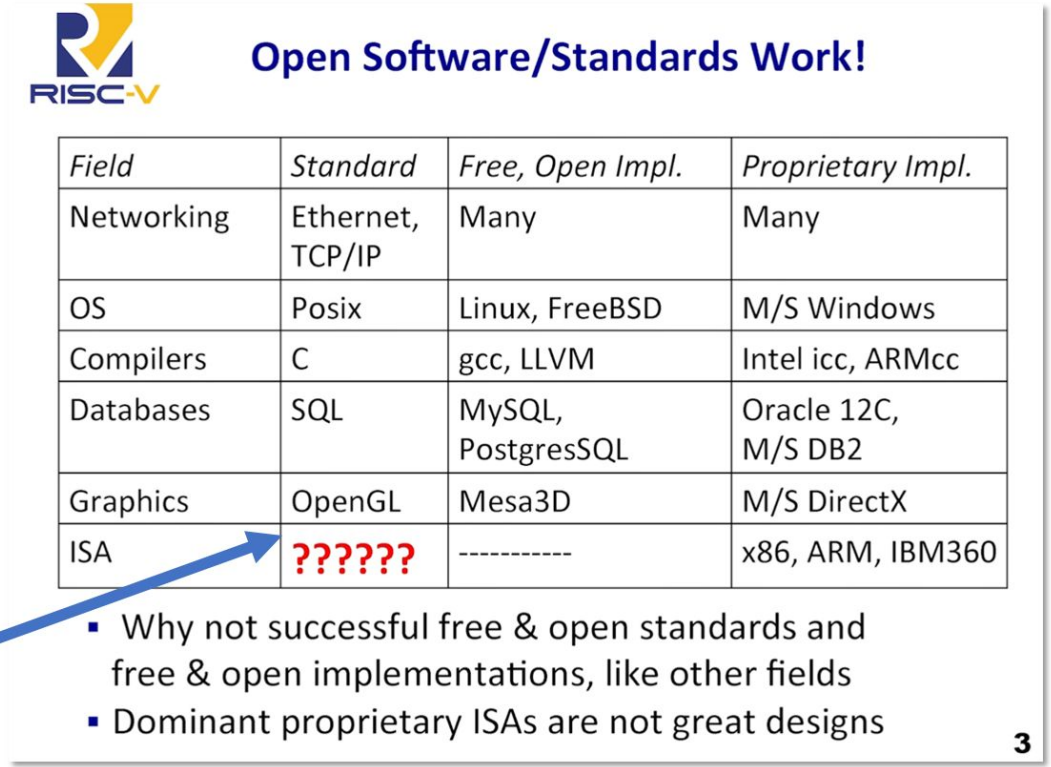
1. Moore's Law has physical limits, but there is no limit to the **demand** for computing power.
2. The **complexity** of a software system grows superlinearly.
3. The number of **developers** capable of managing the complexity of software development is limited.

RISC-V is the future. We're seeing it.

A few conclusions deduced from three basic observations

- Domain Specific Architecture (DSA) will be everywhere.
  - There will inevitably be many (free and open) instruction sets.
- Open source software eats everything.
  - Only a few open source communities can survive in every field.
  - Only a few instruction sets will be maintained by the open source community for a long time with high quality.

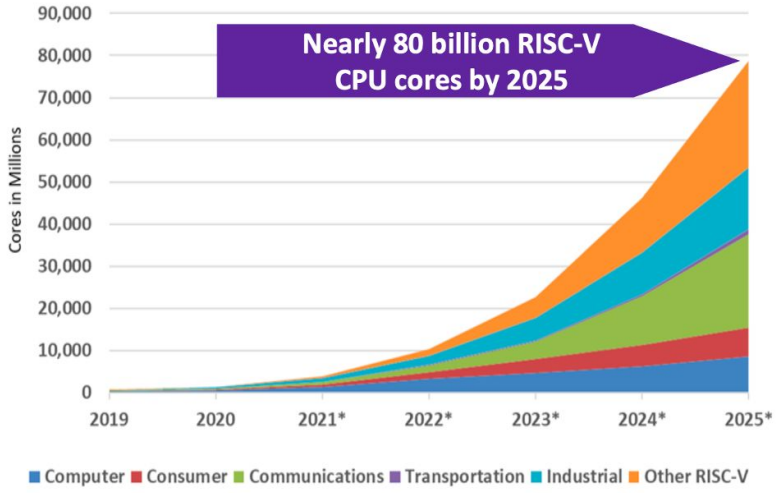
RISC-V is the future. We're seeing it.



"Instruction Sets Want to be Free" Krste Asanovic, Professor of UCB

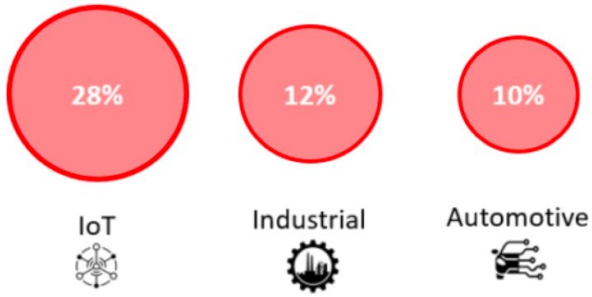
RISC-V is the future. We're seeing it.

RISC-V CPU core market grows 114.9% CAGR, capturing >14% of all CPU cores by 2025



Source: Semico Research Corp, March 2021

RISC-V Penetration Rate by 2025



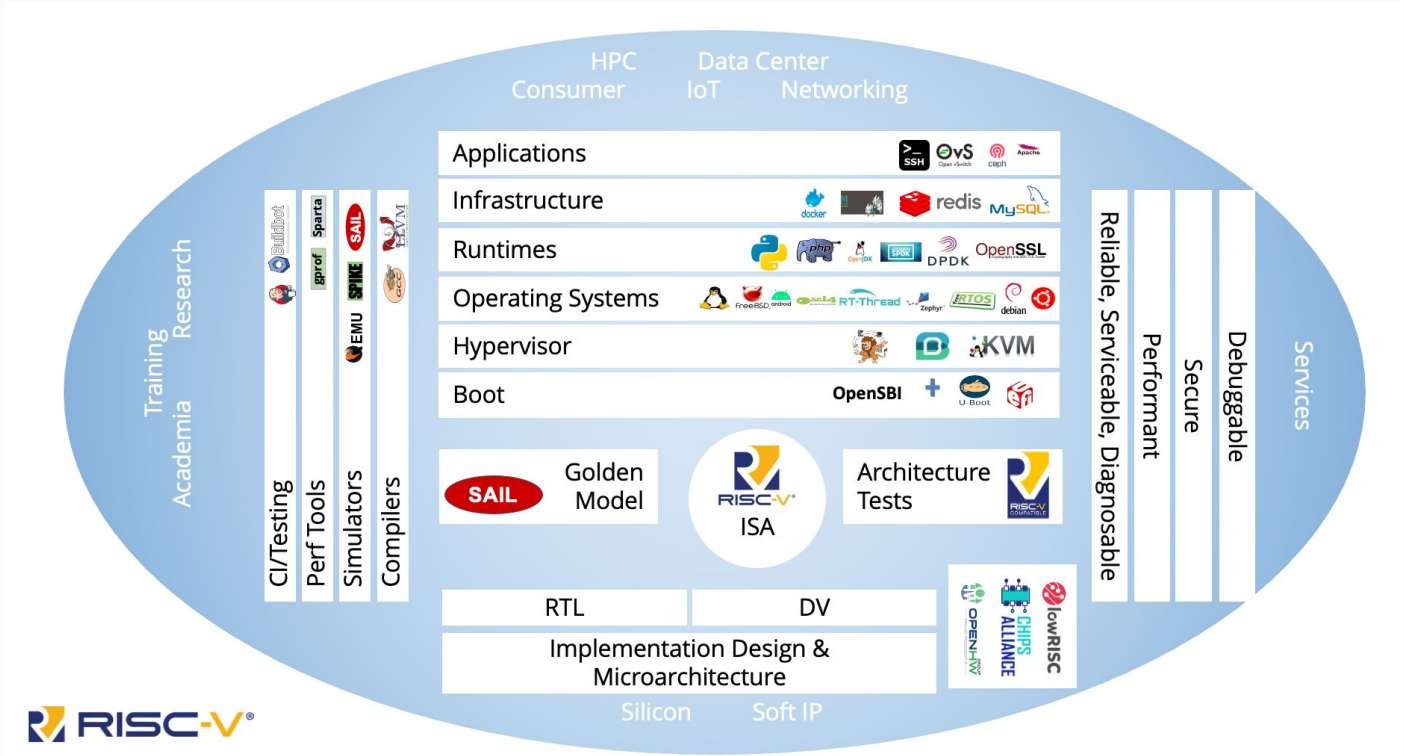
Source: Counterpoint Research, September 2021

"The rise of RISC-V cannot be ignored... RISC-V will shake up the $8.6 Billion semiconductor IP market."

-- William Li, Counterpoint Research

RISC-V is the future. We're seeing it.

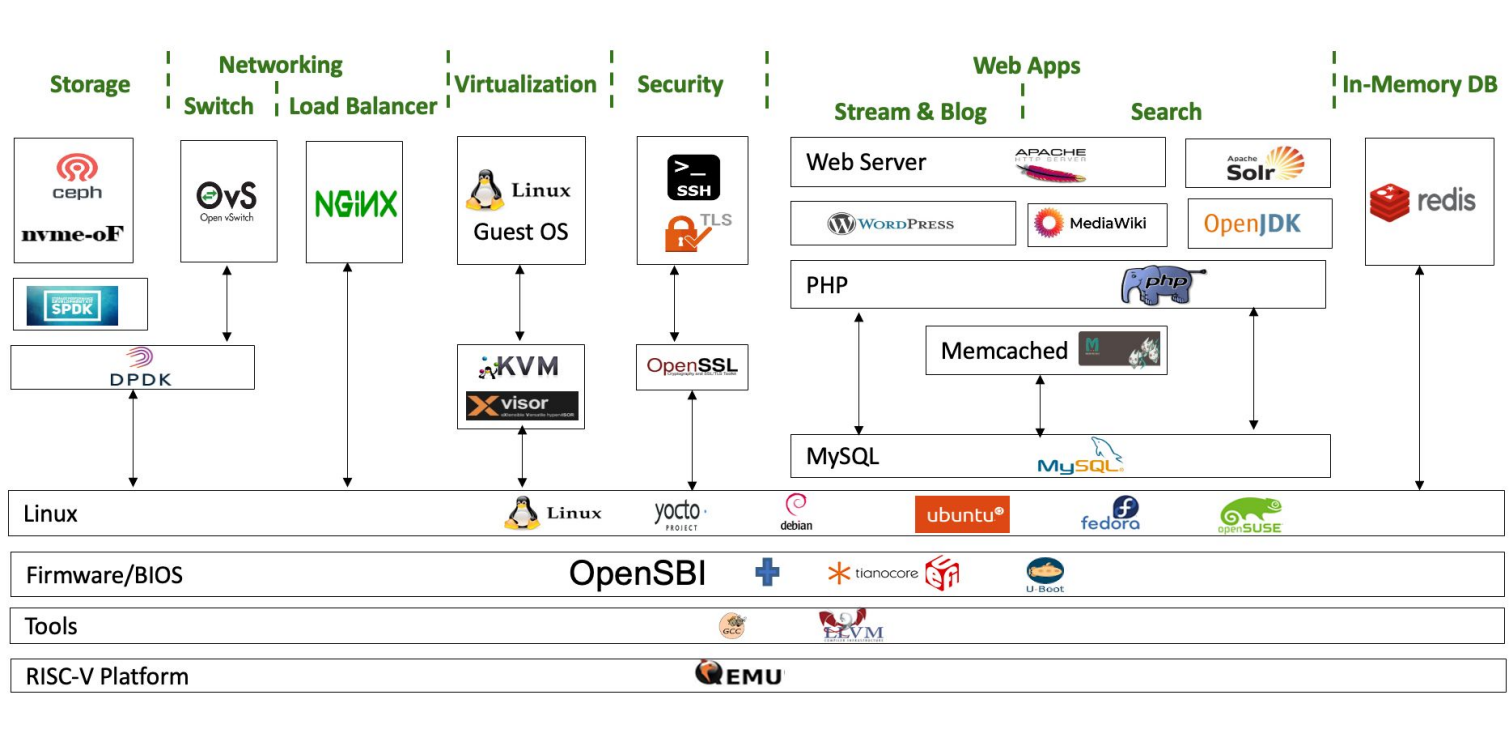
Once you choose RISC-V, you will immediately have all the software tools you need.



Mark Himelstein, https://github.com/cnrv/RVSC2022-Slides/blob/main/Mark-Himelstein-Keynote.pdf

RISC-V is the future. We're seeing it.

Open source software in several mature commercial areas already supports RISC-V.



Mark Himelstein, https://github.com/cnrv/RVSC2022-Slides/blob/main/Mark-Himelstein-Keynote.pdf

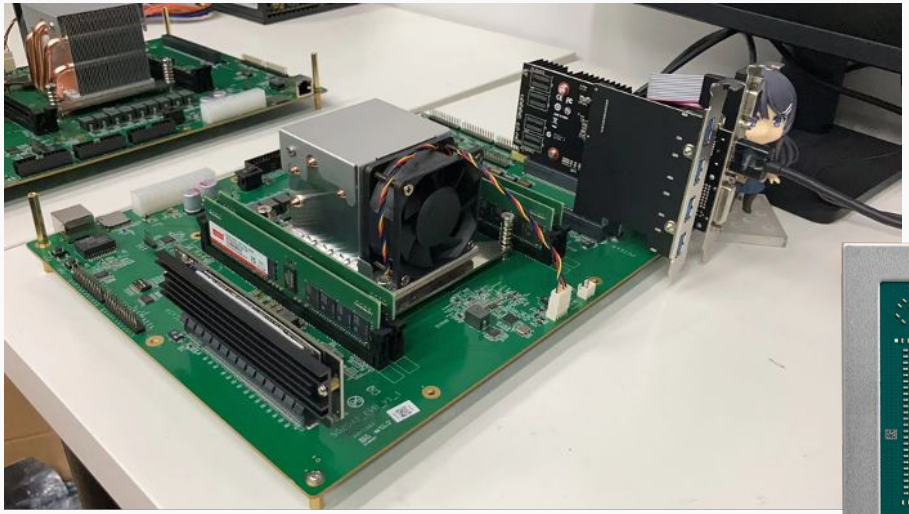
But…
Is RISC-V an Embedded Thing (yet)?

But...
Is RISC-V an Embedded Thing (yet)?

Since 2023, The Answer is NO.

Now we can have SG2042(Workstation/Server) and TH1520(Netbook) in hands.

RISC-V is the future. We're seeing it.



RISC-V is the future. We're seeing it.

Android 12 running on RISC-V with optimised AI performance

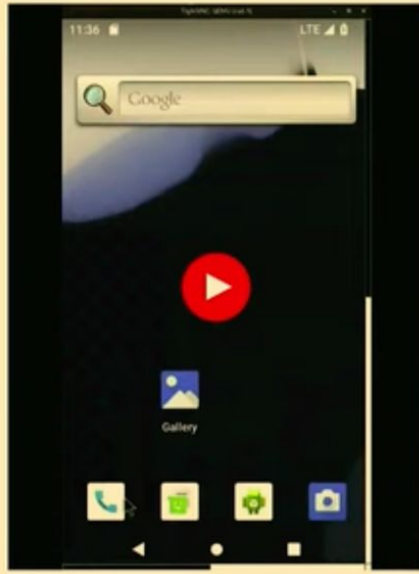
https://chipsalliance.org/announcement/2022/04/21/alibaba-cloud-announced-progress-in-porting-an
https://www.techradar.com/news/alibaba-cloud-is-close-to-getting-android-working-on-risc-v-silicon

Progress in Q1 2023

Emulator with graphics support and ART Java interpreter mode!

Supporting Android and ecosystem adoption of new key RISC-V specifications

- Vector extensions in QEMU, LLVM, and throughout libraries
- Continuing to pick up or extend use of Zbb (bit manipulation) optimizations



Nerds Talking to Nerds About RISC-V (Day-1) →
https://www.bilibili.com/video/BV1z84y1T7Vi/

RISC-V is the future. We're seeing it.

OpenJDK19 will natively support RISC-V

RISC-V compiler support merged on March 24th, 2022

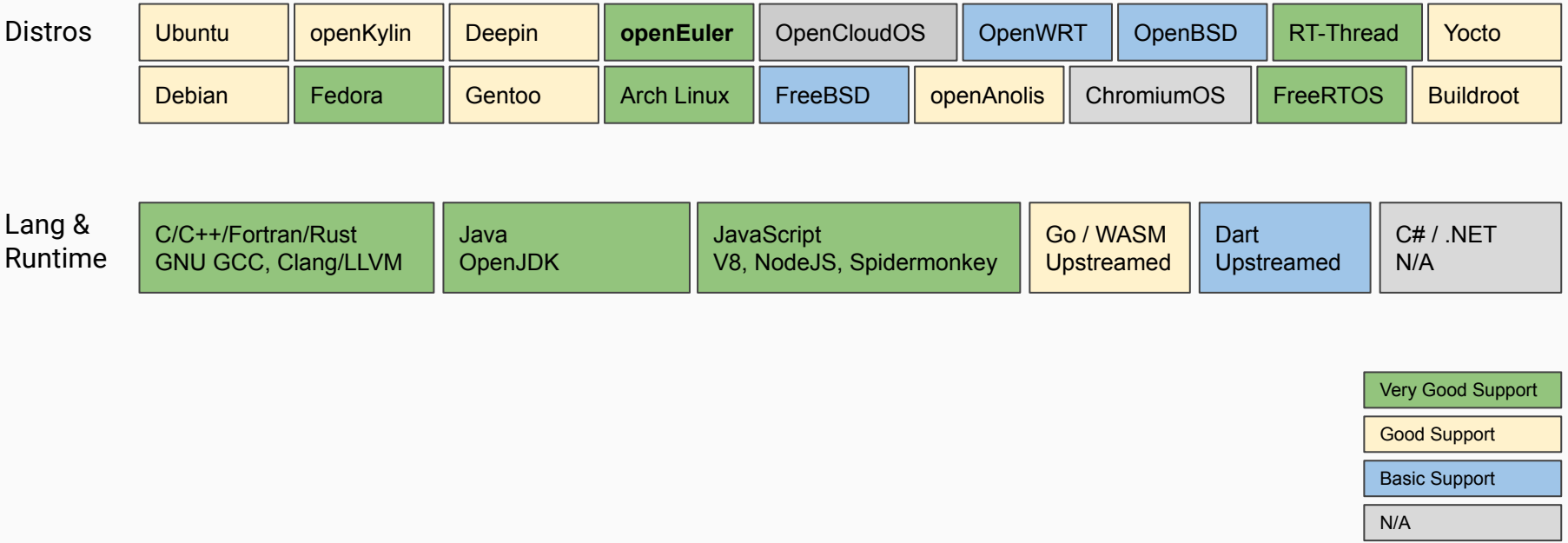
https://github.com/openjdk/jdk/commit/5905b02c0e2643ae8d097562f181953f6c88fc89

17u/11u/8u backport staging repos are ready:
- https://github.com/openjdk/riscv-port-jdk17u
- https://github.com/openjdk/riscv-port-jdk11u
- https://github.com/openjdk/riscv-port-jdk8u

Seeded with jdk{17, 11, 8}u-dev repos respectively.

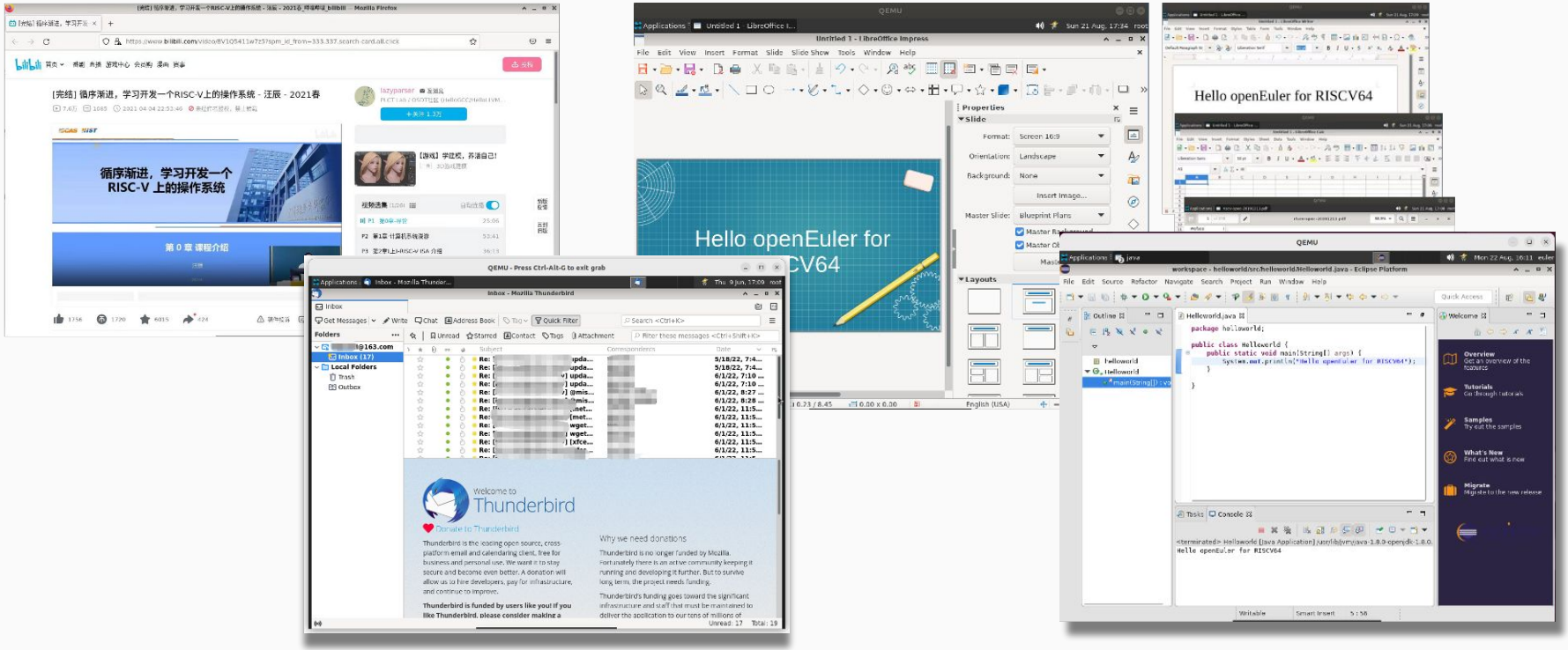
RISC-V is the future. We're seeing it.

In 2022, (almost) all the popular Linux distros support RISC-V. Linux Desktop is almost there.



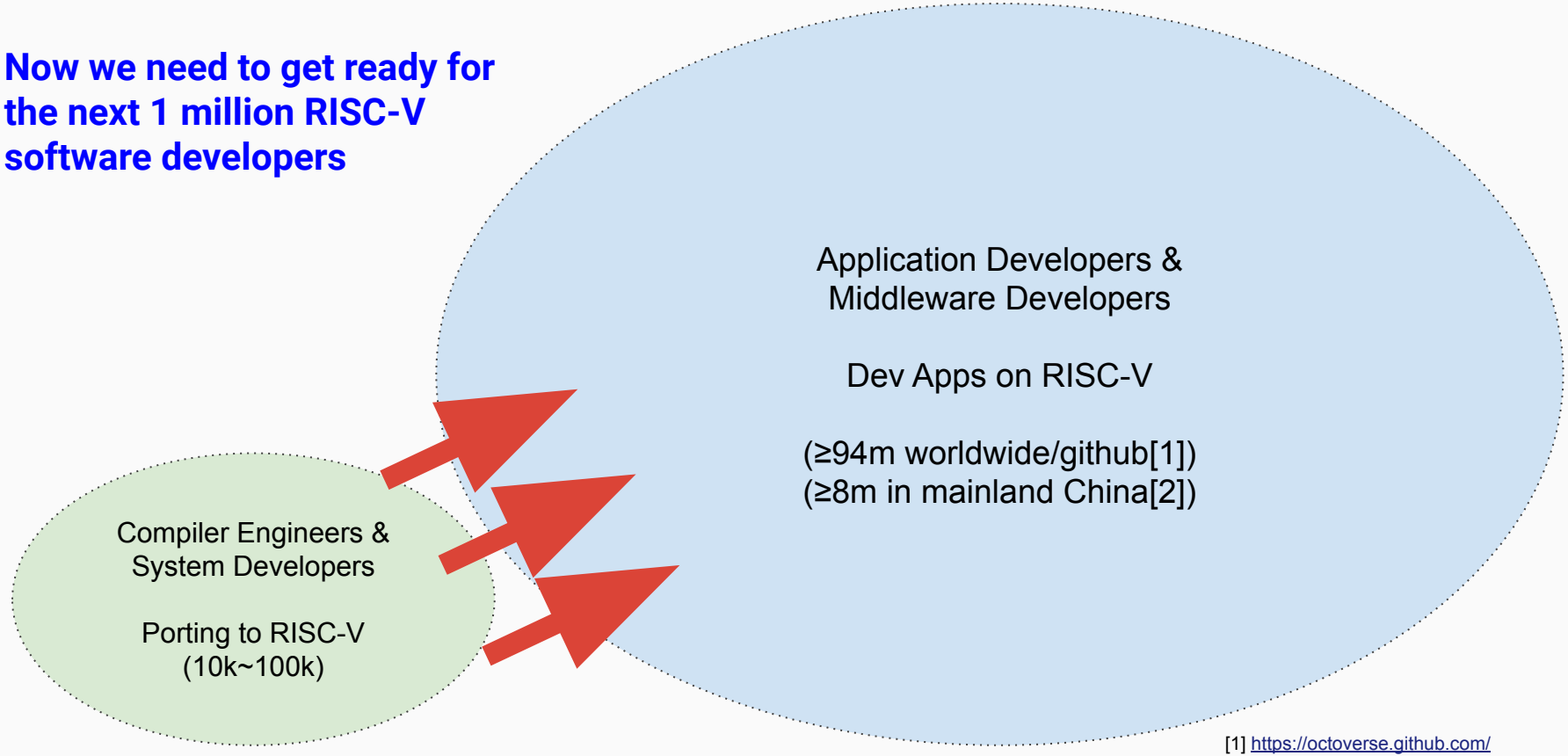
RISC-V is the future. We're seeing it.

Web Browsers (Chromium and Firefox), Mail Clients, Eclipse, LibreOffice, MultiMedia Player, etc.



RISC-V is the future. We're seeing it.

Now we need to get ready for the next 1 million RISC-V software developers



[1] https://octoverse.github.com/
[2] https://zhuanlan.zhihu.com/p/637787110

Why RuyiSDK: The Motivation

The initial idea (and pain) behind RuyiSDK

- The RISC-V instruction set is modular. This is good.
- However, if the user wants to try different combinations,
- So how much does it cost?
- Plus vendor-defined extensions?

The initial idea (and pain) behind RuyiSDK

Back to 2020, PLCT Lab was working on implementing a few ISA extensions like K, B, and V, etc.
We thought it wouldn't be that hard in the beginning…

If a customer wants to try out the effects and pros and cons of different instruction set extension combinations, what should he do now

Source: PLCT OpenDay 2021,
co-located with The 1st RISC-V Summit China

Family Bucket Project: Allow users to test all combinations of instruction set extensions

全家桶计划：让用户可以测试所有指令集扩展的组合

- 如果一个客户想要尝试下不同的指令集扩展组合的效果和利弊，现在他需要怎么做？
  - 例如，要不要加 B 扩展？加 P 扩展还是 V 扩展？
- 源代码 → 编译器 → 模拟器 → FPGA

Source Code → Compiler → Simulator → FPGA

- 特权级 → Boot/Hyper/Kernel → 模拟器 → FPGA

Privilege Mode → Boot/Hyper/Kernel → Simulator → FPGA

当前状态：QEMU 有了公开的分支正在努力；GCC 也正在努力中

Current status: QEMU has a public fork and is working on it; GCC is also working on it

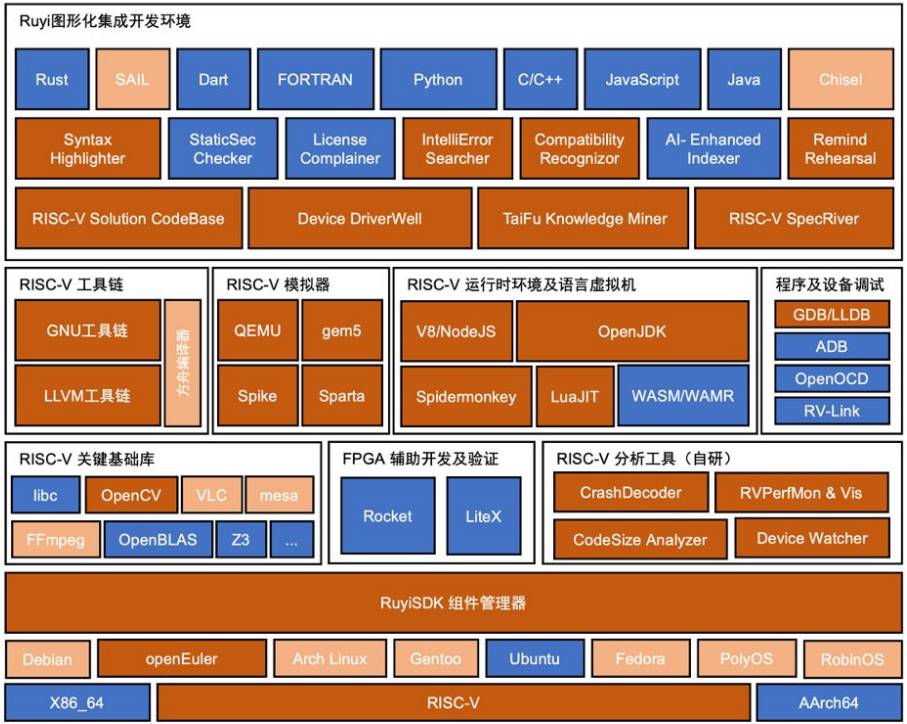
The initial idea (and pain) behind RuyiSDK

- You need to compile the toolchain yourself, and maybe the Kernel, uboot, openSBI, etc.
- Vendors may have closed-source binary toolchains which provides some feature that open source solutions lacked.
  - and "vendor-specific" bugs and crashes
  - There are subtle differences in command-line arguments between different tools, and even between different versions of the same tool.
- Need to distinguish between native compilation and cross-compilation.
- Many downstream or out-of-tree patches.
  - Beware of versions of different system tools and libs!

The Mission of RuyiSDK

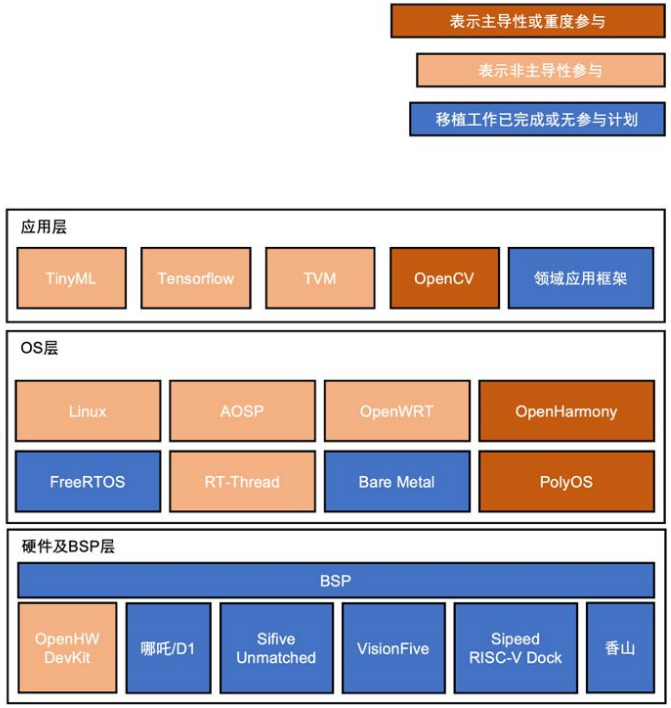
- For non-system developers, it simplifies falling into as many implementation details as possible, and at the same time allows developers to see the principles and processes behind it.
- Provide the same development process, so that users can easily switch between different open source tools and vendor-customized toolchains.
- Simplify the environment construction of cross-compilation and native compilation.
- Provides a set of templates that can start with an existing example

RuyiSDK architecture and components



（开发者环境，原生支持 RISC-V 本地开发）

JTAG等协议



Will support most RISC-V devices available on the market

RuyiSDK

GCC support for RUYISDK

Porting RISC-V extensions on multi-version of GCC, tracking the latest feature and changes for every extension, adapting different Linux OS.

Extension supports: Bitmanip(1.0), Scalar crypto(1.0), Packed SIMD(0.96), Vector(0.7.1 & 1.0) Zfinx(1.0), Zicbo(1.0), Zmmul(1.0), Zc*(1.0), Zfh(1.0), Zfbf(0.1), Profiles(0.9)

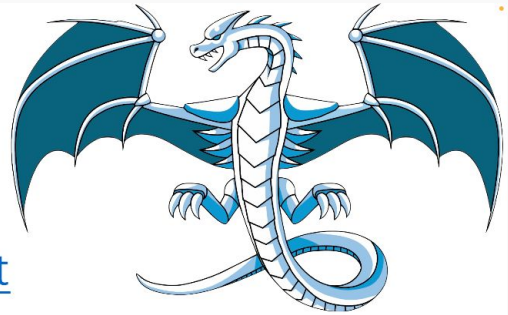
GCC repo: https://github.com/ruyisdk/riscv-gcc
Binutils repo: https://github.com/ruyisdk/riscv-binutils

Developers:

Jiawei-Chen	jiawei@iscas.ac.cn
Shihua-Liao	shihua@iscas.ac.cn
Yulong-Shi	shiyulong@iscas.ac.cn
Yixuan-Chen	chenyixuan@iscas.ac.cn

RuyiSDK architecture and components

Ruyisdk llvm-project



- Upstream https://github.com/llvm/llvm-project
  - RISCV Target
    - Standard extension support for Z*inx, Zc*, Zmmul, scalar cryptography, Zihintpause, Zbpbo
    - Backend Optimization
  - Middle-end Optimization, InstCombine,SCCP,LoopIdiom,LVI
  - Flang
  - LLDB, lldb-server, RV{32,64}I, A M, C, RV32F ,RV64F, D extensions
- CI monitors llvm-project for RISCV regressions every two hours https://lnt.rvperf.org/

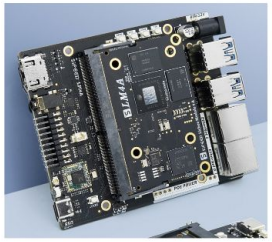
- Downstream https://github.com/ruyisdk/llvm-project
- RVV0.7.1
- Backend Optimization

- Contributors
- LiaoChunyu chunyu@iscas.ac.cn
- luxufan luxufan@iscas.ac.cn
- Shao-Ce SUN sunshaoce@iscas.ac.cn
- Qihan Cai qcai8733@uni.sydney.edu.au
- Yingwei Zheng dtcxzyw2333@gmail.com
- melonedo funanzeng@gmail.com
- vincentWu vincenttttwu@gmail.com
- Emmmer yjhdandan@163.com

RUYI QEMU

- Works
  - Port support for Xuantie CPUs
    - Support for RVV 0.7.1 and P extensions
    - Support for Xuantie specific vector instructions
    - Support BF16 format for float point related intructions
  - Initial support for Lichee Pi RV and Lichee Pi 4A machines

T-HEAD



- Links: https://github.com/plctlab/plct-qemu/tree/ruyi-qemu

- Main Contributor: Weiwei Li <liweiwei@iscas.ac.cn>

(WIP) ruyi-ng: Next Gen Distro/DevEnv Manager

- RISC-V distro current status
  - Tons of rootfs (rv64gc, rv64gcv, rv64gcv0p7, etc)
  - Tons of repositories, rolling (Arch, Fedora, RevyOS, etc)
- People use a distro for reasons
  - Not for flame war!
  - Seems alike, but tons of difference when actual dev
  - Not try to eliminate them, include them!
- Dev env essentials
  - Anyone can reproduce
  - Minimal setup steps
  - One-click to distribute
  - RISC-V on x86: binfmt trick
- Reproducible: Nix mimic, but only server side
  - Base Images from known groundtruth
  - Hash-locked Software Repo[a]
- Handy CLI: Mix of conda and docker semantics[b]
  - `ruyi-ng checkout archrv-monthly archrv-with-todays-rust`
  - `ruyi-ng activate archrv-with-todays-rust`
  - Install rust nightly, no back-and-forth Dockerfile
  - `ruyi-ng commit`
  - `ruyi-ng publish`

[a] WIP: `https://github.com/NickCao/nap`
[b] WIP: `https://github.com/NickCao/ruyi-ng`

(WIP) ruyi-ng Demo: Dev on two distro

```
[zenithal@Aldebaran ruyi-ng]$ ruyi-ng activate archrv-gcc
[root@Aldebaran ~]# neofetch
                    -`                    zenithal@Aldebaran
                   .o+`                   ------------------
                  `ooo/                   OS: Arch Linux riscv64
                 `+oooo:                  Kernel: 6.1.11
                `+oooooo:                 Uptime: 21 mins
                -+oooooo+:                Packages: 119 (pacman)
              `/:-:++oooo+:               Shell: bash 5.1.16
             `/++++/+++++++:              Terminal: /dev/pts/1
            `/++++++++++++++:             CPU: (64)
           `/+++ooooooooooooo/`           Memory: 267MiB / 64166MiB
          ./ooosssso++osssssso+`
         .oossssso-````/ossssss+`
        -osssssso.      :ssssssso.
       :osssssss/        osssso+++.
      /ossssssss/        +ssssooo/-
    `/ossssso+/:-        -:/+osssso+-
   `+sso+:-`                 `.-/+oso:
  `++:.                           `-/+/
  .`                                 `/

[root@Aldebaran ~]# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
Target: riscv64-unknown-linux-gnu
```

Figure: Arch RISC-V with GCC

```
[zenithal@Aldebaran ruyi-ng]$ ruyi-ng activate revyos-xthead-gcc10
[root@Aldebaran ~]# gcc -v |& grep -o 'with-arch=\w*'
with-arch=rv64imafdcv0p7_zfh_xtheadc
[root@Aldebaran ~]# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

Figure: RevyOS with T-Head GCC

Beyond RuyiSDK

The Next "Big Project"

≥ 100 chip & OS companies | ≥ 200 app comanies | ≥18 field | ≤5 years

Questions?