Unchain Your Toolchains with CROPS (CROssPlatformS)

Todor Minchev
todor.minchev@linux.intel.com
Intel Open Source Technology Center

Embedded Linux Conference
San Diego, 5 April 2016

intel Software

Intel OpenSource TECHNOLOGY CENTER

Agenda

- Traditional Cross-Platform Development Workflow
- CROPS Definition & Value
- Native Tools vs CROPS Containers
- Technical Overview
- Current Status
- Future Plans & Challenges
- Demo
- Q & A

Traditional cross-platform development workflow

GDB
The GNU Project Debugger

Traditional cross-platform development workflow on multiple host platforms

Linux

Linux

GDB
The GNU Project Debugger

Windows

Windows

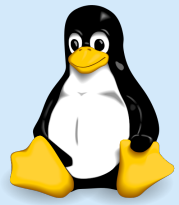
GDB
The GNU Project Debugger

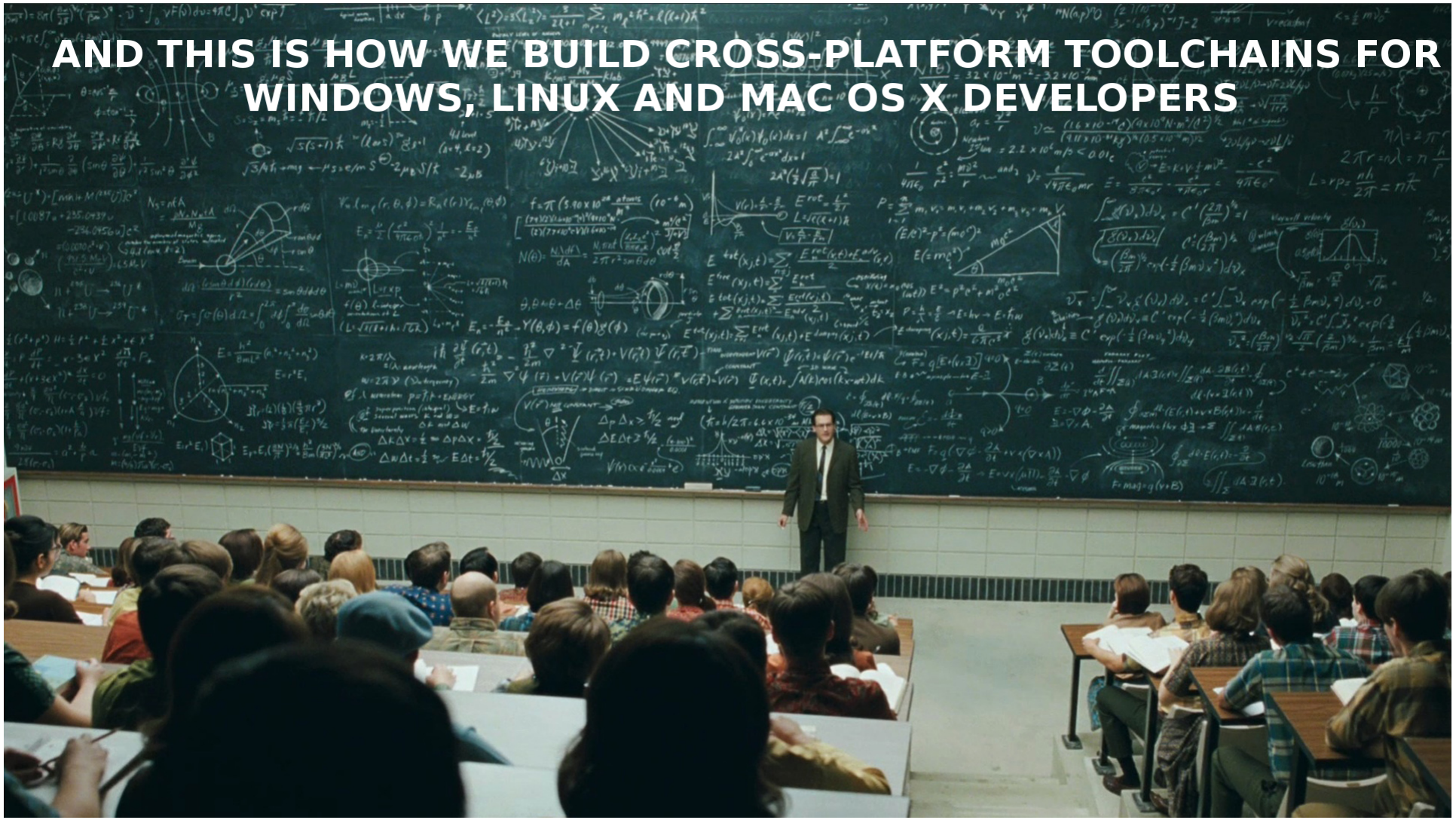
Mac OS X

Mac OS X

GDB
The GNU Project Debugger

Scalability of the traditional cross-platform development model

DEVELOPMENT HOST (N)	SDK (N x M)	TARGET PLATFORMS (M)
	X86	X86
	ARM	
	MIPS	
	OTHERS	ARM
Windows	X86	
	ARM	
	MIPS	MIPS
	OTHERS	
Mac OS X	X86	
	ARM	OTHERS
	MIPS	
	OTHERS	


AND THIS IS HOW WE BUILD CROSS-PLATFORM TOOLCHAINS FOR WINDOWS, LINUX AND MAC OS X DEVELOPERS

What is CROPS?

CROPS is an open source, cross-platform development framework that leverages Docker containers to provide an easily managed, extensible environment which allows developers to build binaries for a variety of architectures and use native Linux tools on Windows, Mac OS X and Linux hosts.

What Value does CROPS provide?

CROPS provides the following capabilities:

- a solution to allow cross building for different targets from Windows, Mac, & Linux hosts
- the ability to leverage Linux based tools in addition to the cross compiler e.g. bitbake, image creator, kernel menuconfig, perf, oprofile
- a path to embrace the cloud as part of the solution
- an alternative to a full Linux VM
- easy toolchain distribution and updates
- a clean, reproducible state for development and testing

Native Solution

Port the Linux toolchains for all the desired architectures to each of the desired hosts

PROS		CONS
No added complexity from containers		Need to qualify N toolchains on M hosts
Based on well known technologies		Windows represents several hosts (7,8,8.1,10, future…)
Meets user expectation for a toolchain app		Toolchain updates are likely monolithic, making rollbacks harder
		Easily polluted by host environment
		Remote builds infeasible

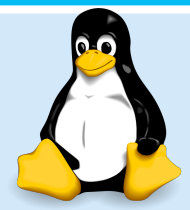
Containers Solution

Leverage containers to solve problem needs while relying on the host for the graphical IDE

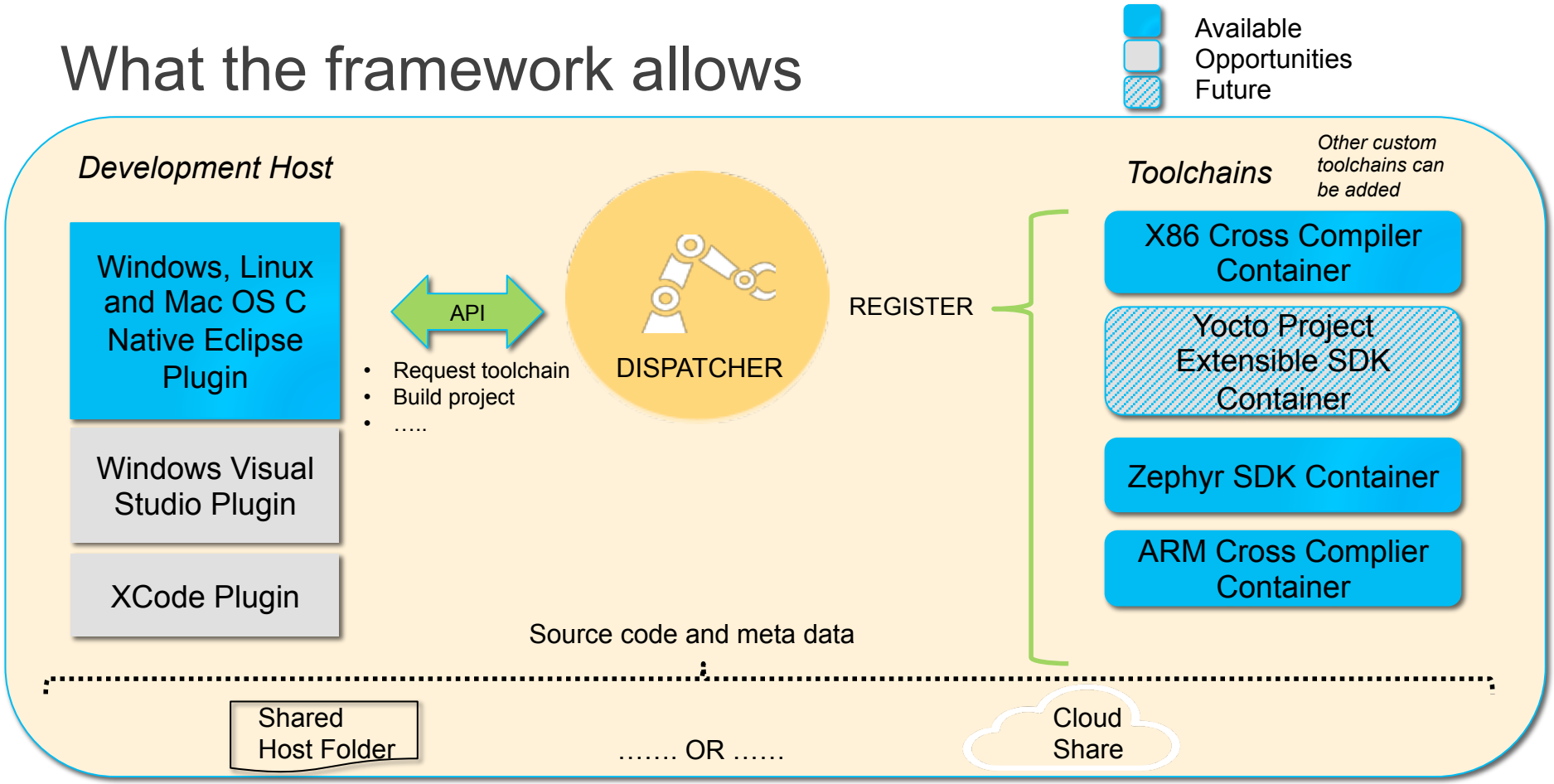
PROS	CONS
Need to qualify N toolchains on only 1 container	Containers add additional level of complexity (largely hidden)
Easy toolchain distribution, updates, and rollbacks	Based on forward-looking, state of the art technologies
Isolated from host environment	
Supports remote builds	
Active community support enabling containers on different hosts	

The primary advantage of containers is that we concentrate on the API and what is inside the container while others manage the issue of making it work on diverse platforms.

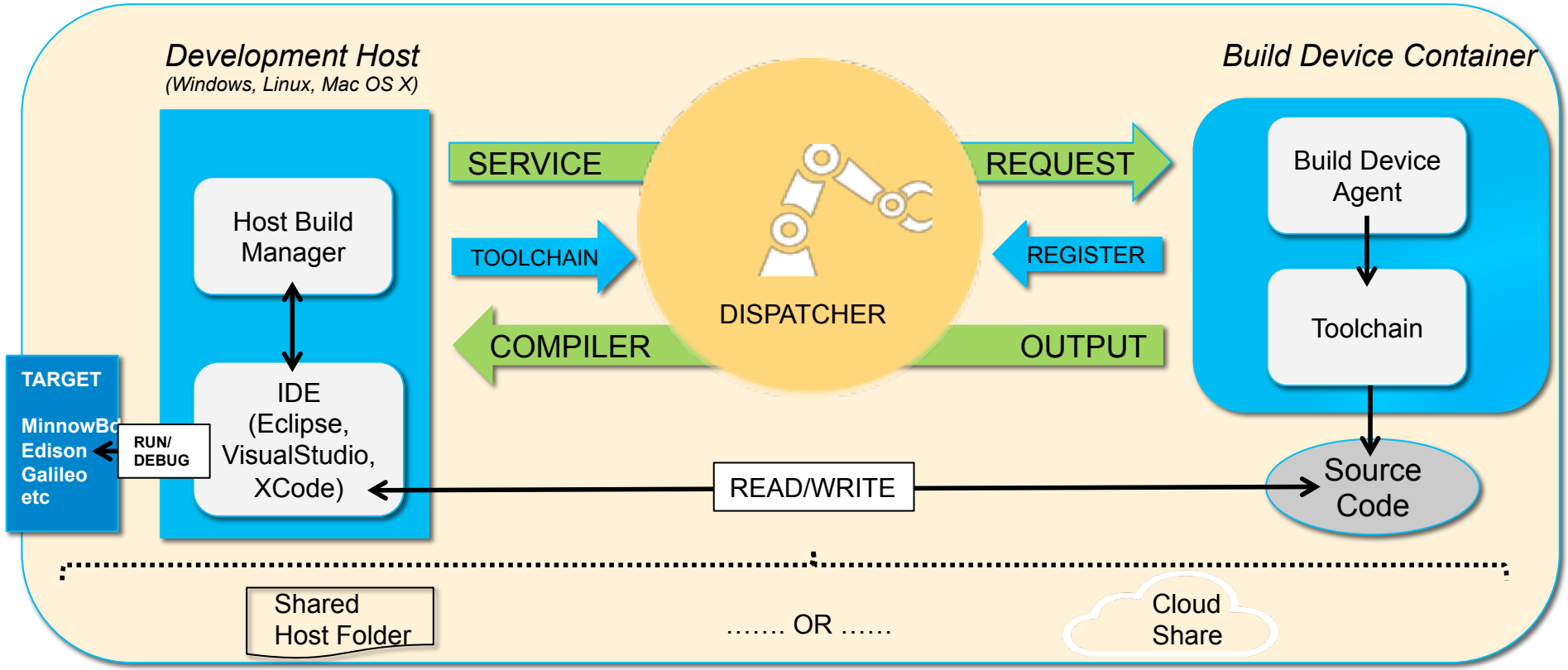
Scalability of the CROPS development model

DEVELOPMENT HOST (N)	CROPS SDK(M)	TARGET PLATFORMS (M)
	X86	X86
Windows	ARM	ARM
Mac OS X	MIPS	MIPS
	OTHERS	OTHERS

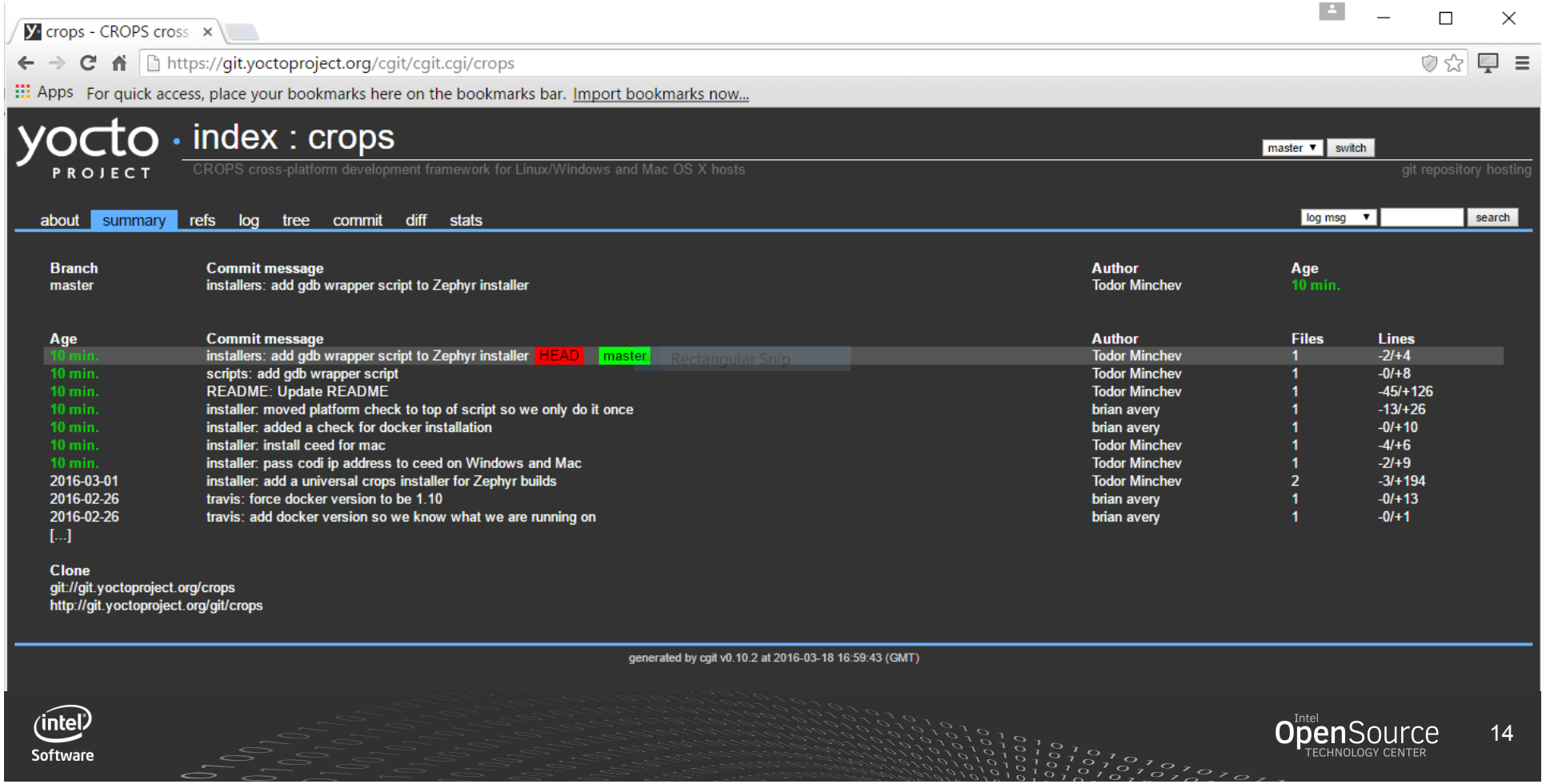
What the framework allows



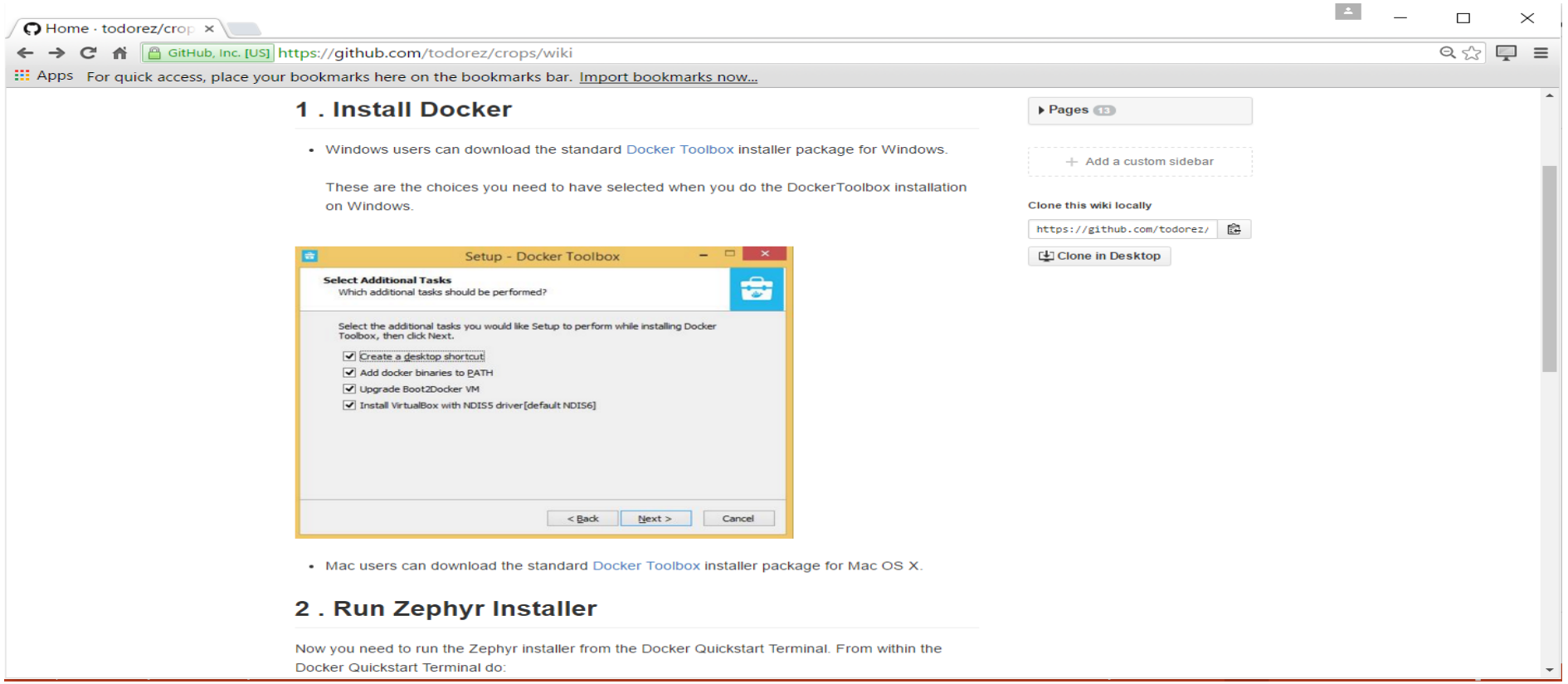
How CROPS works



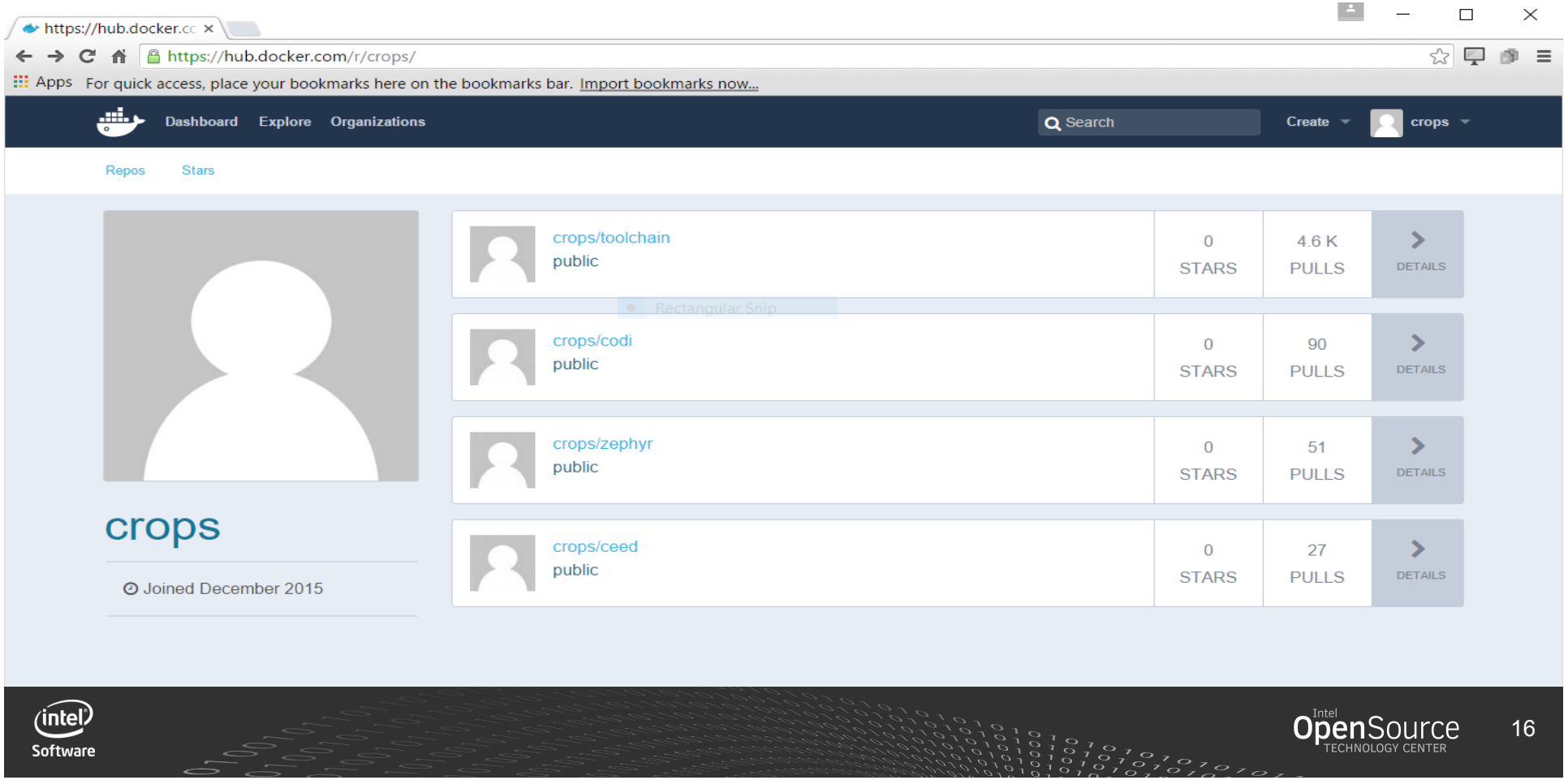
Current Status – Project Home (https://git.yoctoproject.org/cgit/cgit.cgi/crops/)



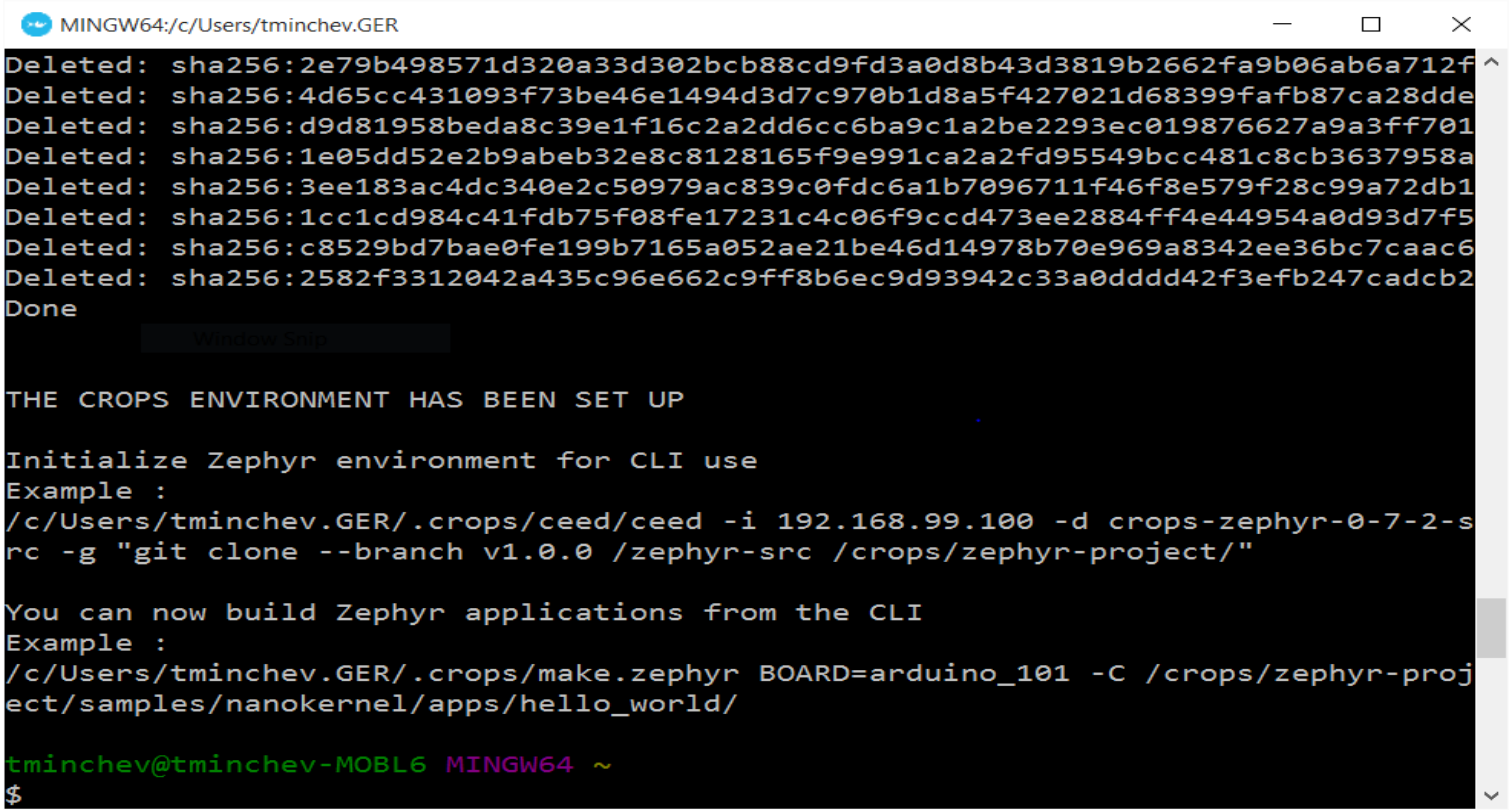
Current Status - GitHub mirror with wiki (https://github.com/todorez/crops)



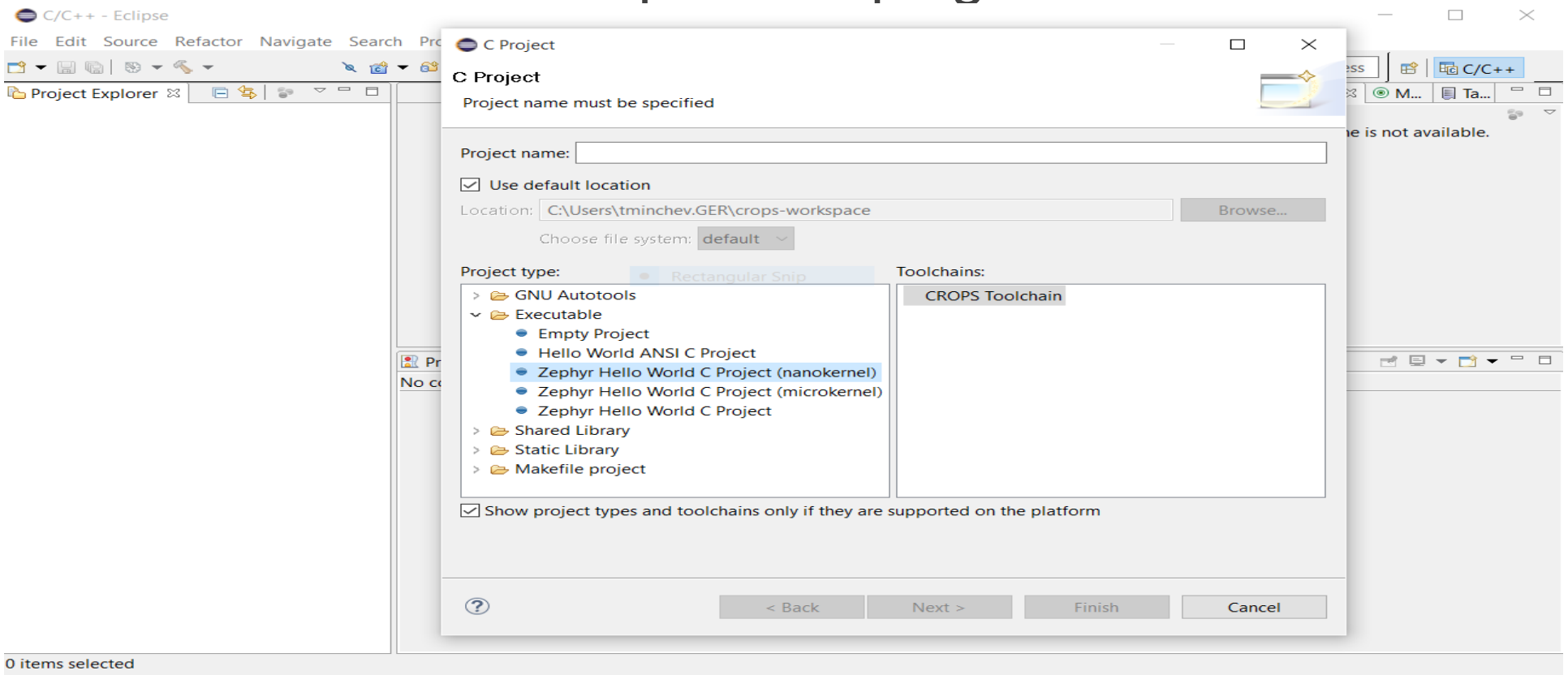
Current Status – DockerHub (https://hub.docker.com/r/crops/)



Current Status - CLI



Current Status – Eclipse IDE plug-in



Future Plans

- **RESTful API**
  - Current framework uses Internet sockets
  - Firewalls allow only well known ports through (e.g 80, 443)
- **Remote Toolchain/Projects Support**
  - Host toolchain containers remotely
  - Store project workspaces remotely
  - Share toolchains
- **Toolchain descriptors**
  - Describe toolchain capabilities
  - Supported architectures
  - Default compiler flags

Future Plans

- **Dynamic Eclipse IDE UI**
  - Provide different UI perspectives based on toolchain capabilities
  - RTOS builds vs userspace application builds

Challenges

- **Remote Projects Support**
  - File synchronization
  - Depends on Internet connectivity
  - Binary File Diffs

- **Debugging from Eclipse on Windows**
  - Pseudo terminals on Windows

DEMO

Q & A

Intel Software

Intel OpenSource TECHNOLOGY CENTER

CROPS Team

Todor Minchev
todor.minchev@linux.intel.com

Brian Avery
brian.avery@intel.com

Tim Orling
timothy.t.orling@intel.com

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

- Other names and brands may be claimed as the property of others.

intel Software

Intel OpenSource TECHNOLOGY CENTER