



arm

CMSIS v6

CLI Builds using CMSIS-Toolbox

Hans Schneeberger, Chen Tang – Arm

Rafael Taubinger – IAR Systems

13 February 2024

CMSIS-Toolbox: command-line tools to work with CMSIS-Packs

Build embedded projects on Desktop or Cloud using Windows, Linux or MacOS

What?

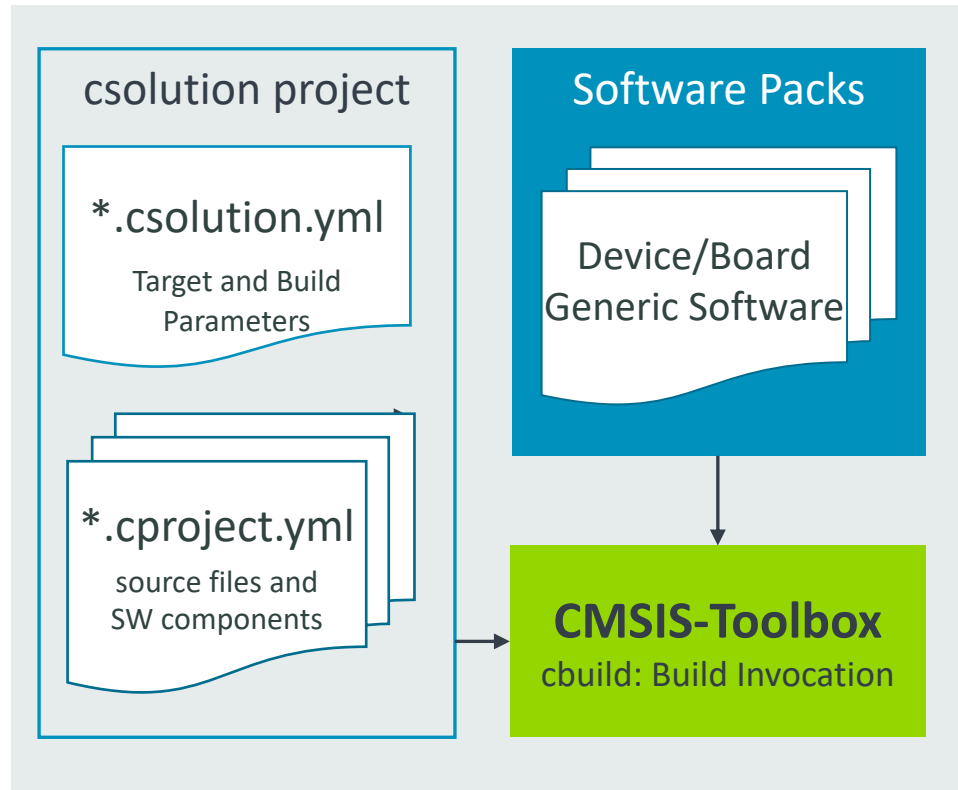
- + CMSIS solutions (*csolutions*) are a collection of projects that are independently managed. The operation is controlled via intuitive [csolution project files in YAML format](#).
- + Uses CMSIS software packs for device/board support and access reusable software components.
- + Product lifecycle management (PLM) with versioned software packs and management for configuration files.

Why?

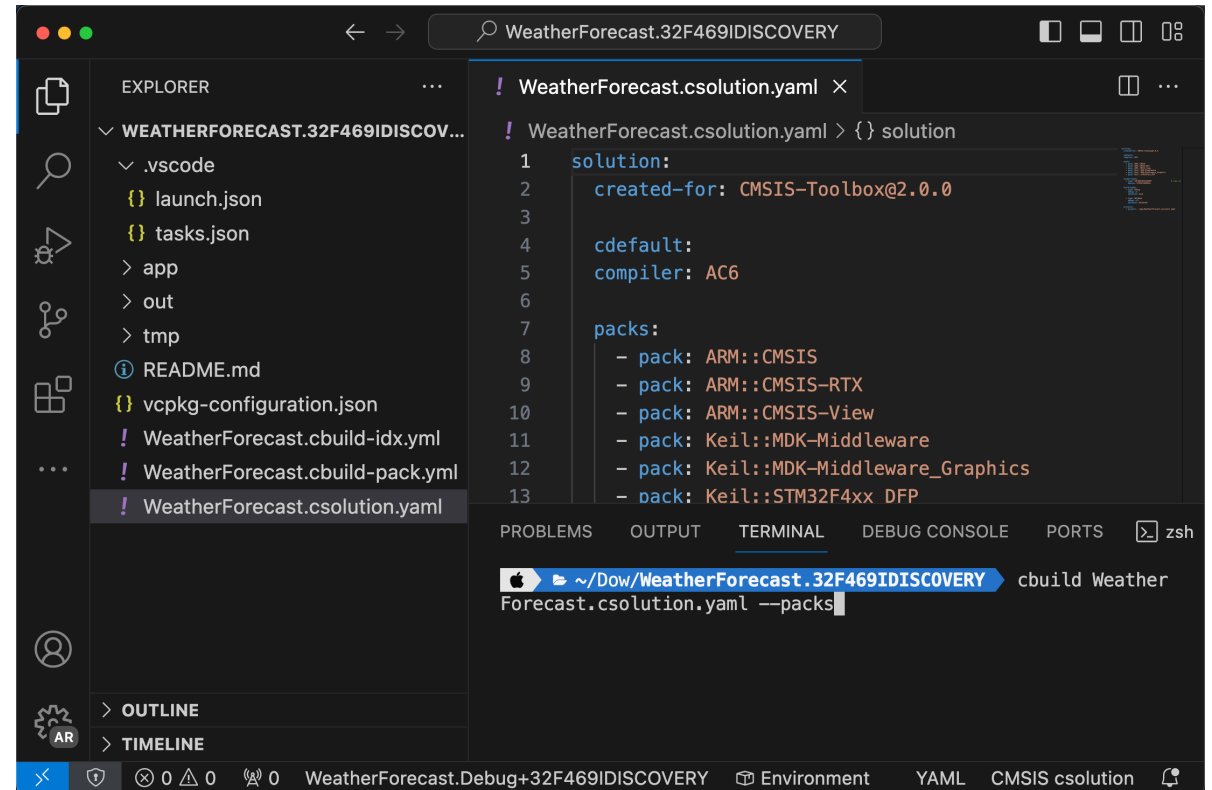
- + Embedded applications are divers and need multiple projects parts, for example for multi-core, boot loader, or unit test for validation.
- + Simplifies tool configuration and gives you the software required to start projects quickly.
- + Reproducible builds and keeping track of third-party components is important as embedded applications have a long live span.

Basis for next generation software tooling

Command line workflow

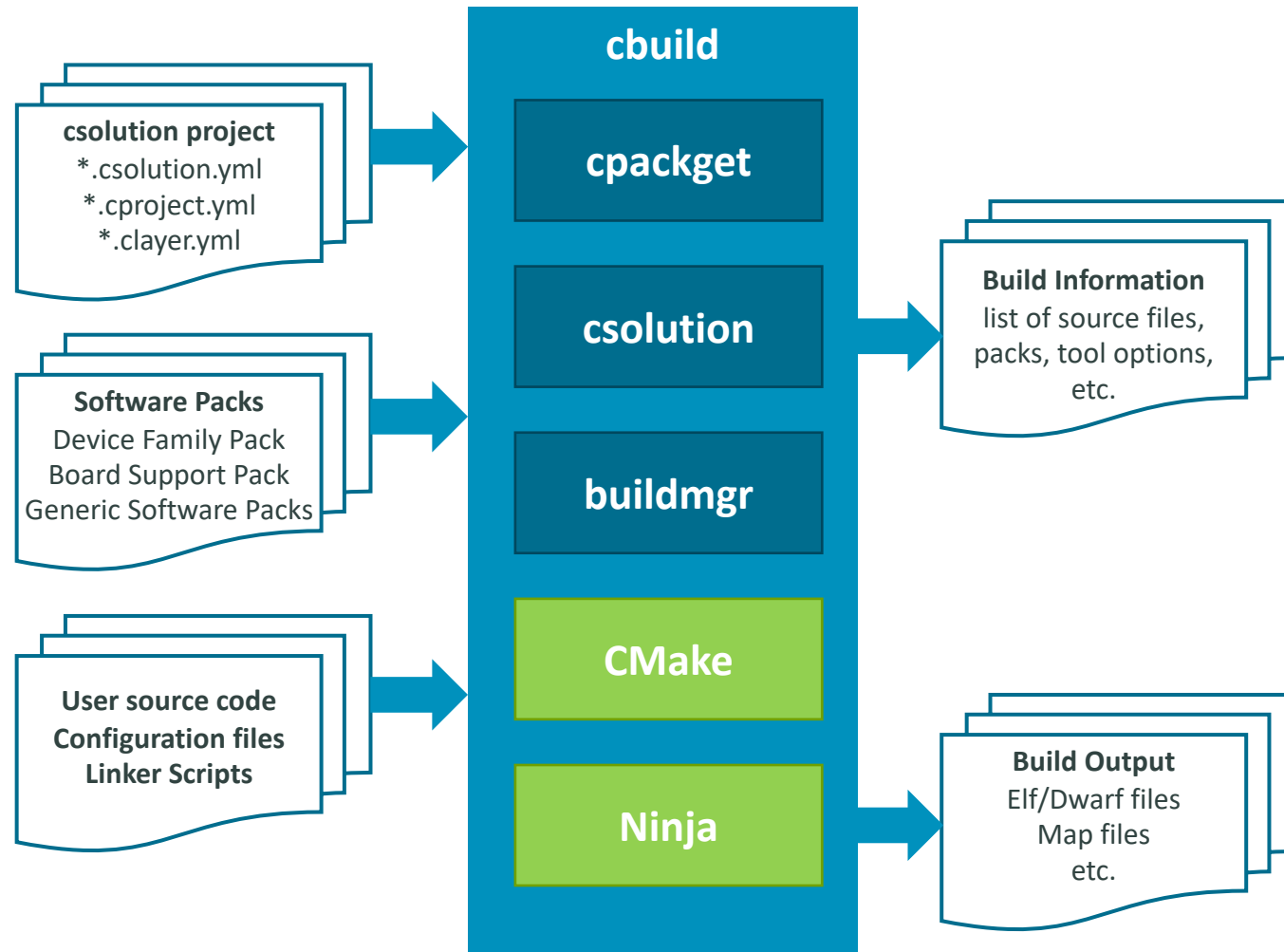


Visual Studio Code IDE



github.com/Open-CMSIS-Pack/csolution-examples

Overall Workflow



arm

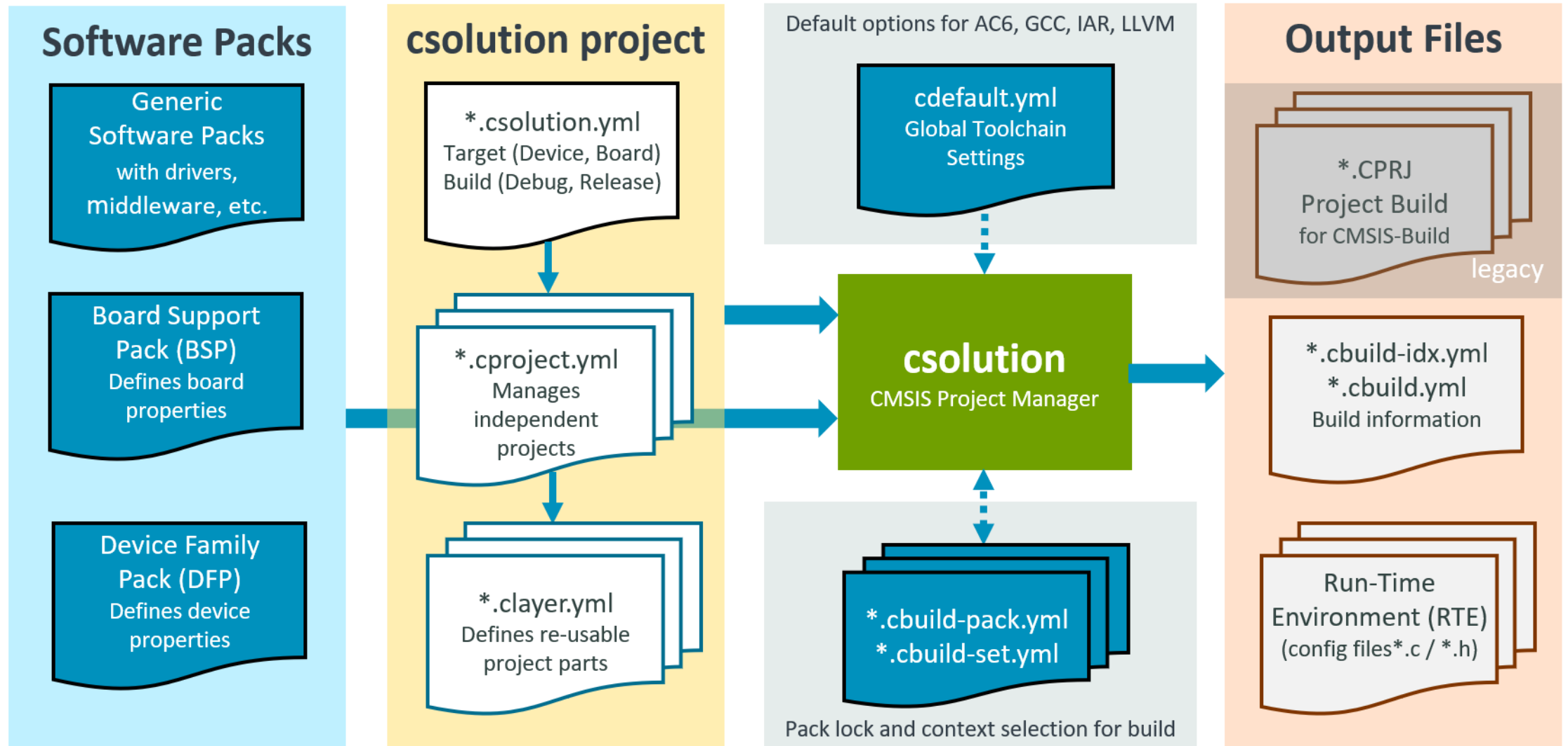
Demo

Using the CMSIS-Toolbox

Manual Chapters

- + [Overview](#) outlines the CMSIS-Toolbox.
- + [Installation](#) explains the setup of the CMSIS-Toolbox.
- + [Build Overview](#) describes the overall concept, outlines the csolution project files that describes the software application, and contains references to examples and templates.
- + [Build Tools](#) describes the command line of cbuilder, cpackget, and csolution.
- + [CMSIS Solution Project File Format](#) delivers a detailed description of YAML format of the csolution project files.
- + [Build Information Files](#) covers details about the build information generated (contains bill-of-material including license information).
- + [Build Operation](#) contains further details for advanced users.
- + [Pack Creation](#) describes how to create software packs.

Operation of the csolution CMSIS Project Manager



Multi-project features in CMSIS-Toolbox

A [context](#) identifies a configuration with `<project-name>.<build-type>+<target-type>`

- + `<project-name>` – a *csolution* combines related projects that are generated independently
- + `.<build-type>` – tool specific setups such as *Debug*, *Release*, or *Test*
- + `+<target-type>` – multiple hardware targets: *evaluation board* or *production hardware*
- + [context-set](#) – build applications with different context types, for example:

```
cbuild iot-product.csolution.yml -c TFM.Release+Board -c MQTT_AWS.Debug+Board -c Bootloader.Release+Board -S
```

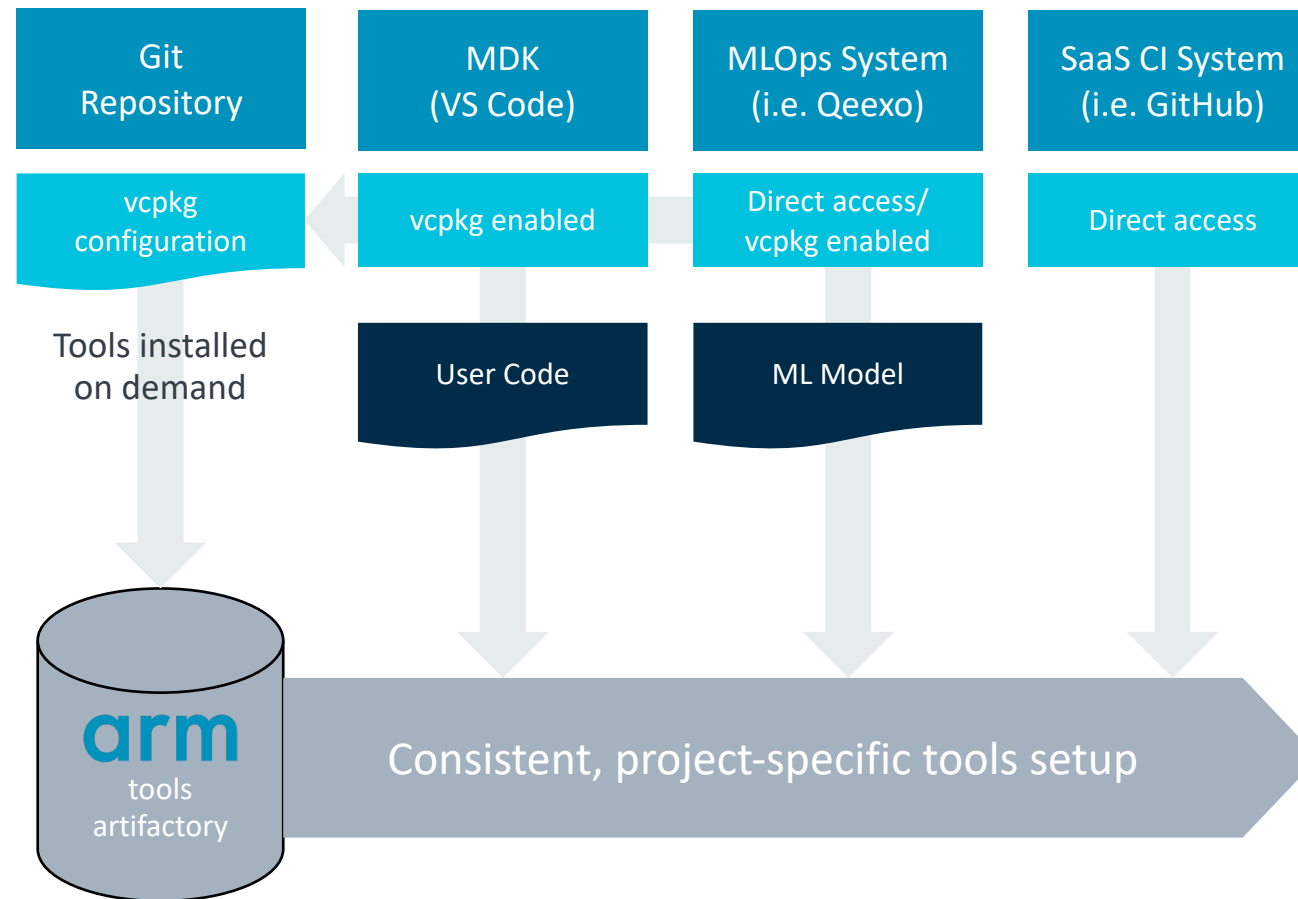
arm

Demo

Using multiple projects

Automated delivery of Arm tools

Tool deployment to MDK (VS Code), CI SaaS, and MLOps



- + artifacts.tools.arm.com provides access to all tools for installation in different environments.
- + Microsoft vcpkg simplifies the tool installation across various host systems.
 - The `vcpkg_configuration.json` file specifies the required tools.
 - Adding `vcpkg_configuration.json` to the project ensures consistent setup.
- + MLOps and CI systems may access tools directly.
- + Example for Docker setup: github.com/ARM-software/AVH-MLOps

Integration with Tools

- ✦ [Linker Script Management](#) – define the available memory resources and memory map for each project context.
- ✦ [Generator Support](#) – for tools such as CubeMX or MCUXpresso.
- ✦ [VS Code Integration](#) – IDE integration with views to your project context set.
- ✦ [μVision Integration](#) – export projects from MDK v5, modify/build csolution projects in μVision or use the μVision debugger.
- ✦ [IAR EW-Arm Integration](#) – use the features of IAR with csolution projects.

arm

Demo

Using CMSIS-Toolbox with μ Vision

arm

Demo

Using CMSIS-Toolbox with IAR EWARM

Upcoming webinar: CMSIS-Stream and SDS

- + CMSIS-Stream provides methods, interfaces, and tools for data block streaming between processing steps of a DSP/ML application.
- + SDS implements a data stream management, provides methods and helper tools for DSP and ML projects.

February 27th

arm

Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكراً

ধন্যবাদ

תודה

ధన్యవాదములు