Symphony No. 42 in C#, Op. 1, Conducted by dapr Workflow

Edwin van Wijk

Principal Solution Architect / Managing consultant

Info Support Solid Innovator

Microsoft Most Valuable Professional

Dapr building blocks

Service-to-service invocation	State management	Publish and subscribe	Resource bindings and triggers	Actors	Secrets	Configuration	Distributed Lock	Workflow	Cryptography
Perform direct, secure service-to-service method calls	Store key-value data containing the state of long running services	Secure and scalable messaging between services	Trigger code based on events from many different inputs Publish events to many different outputs	Encapsulate code and data in stateful virtual actors	Securely access and use secrets from your application	Retrieve and subscribe to application configuration items from configuration stores	Synchronize access to shared resources in a distributed system	Automate and orchestrate tasks within your application	Perform operations for encrypting and decrypting data

For this session, I assume you already have basic understanding of what dapr is!

I also assume you're not here for a talk about Dapper !

Dapr Workflow

- Building block for building process logic (sagas) for orchestrating microservices in a reliable way

- Workflows are stateful and can be used for building long-running and fault-tolerant applications

- Workflows can be managed be used using a Dapr HTTP/gRPC API
  - SDKs exist for several programming languages

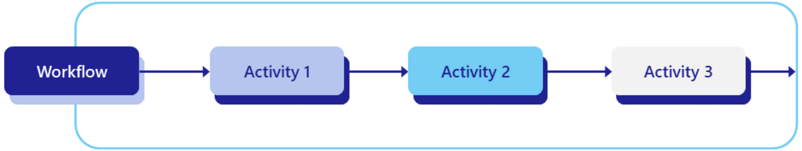
- Currently in beta!

Dapr Workflow

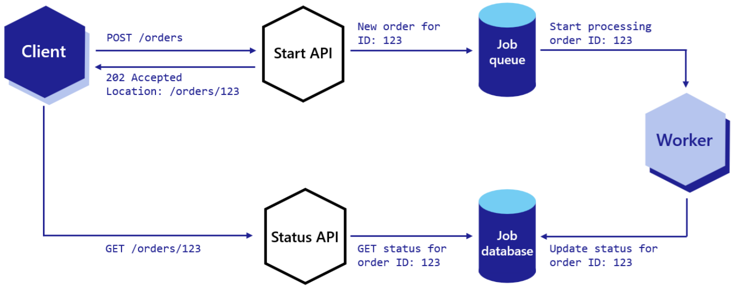
- Workflows can work seamlessly with other Dapr building blocks
- Implemented using the using the Dapr Actor Model building block
- Supports
  - Workflows and activities
  - Calling child workflows
  - Durable timers & reminders
  - Retry policies
  - Waiting for external events that are raised by external systems
  - Distributed tracing using Open Telemetry

Workflow patterns

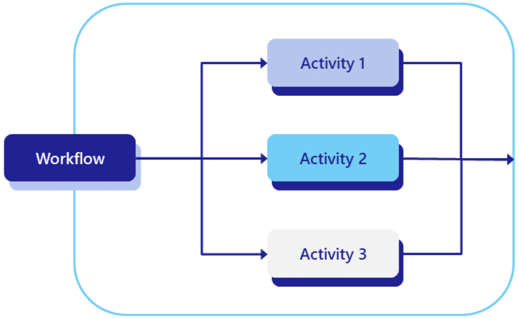
Task Chaining



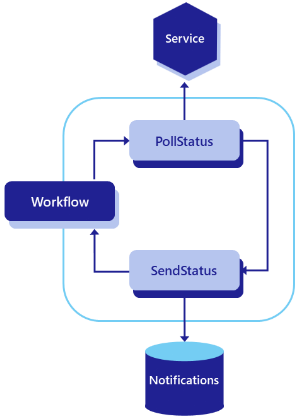
Async API



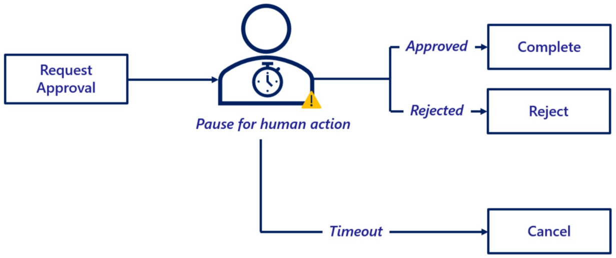
Fan-out / Fan-in



Monitor

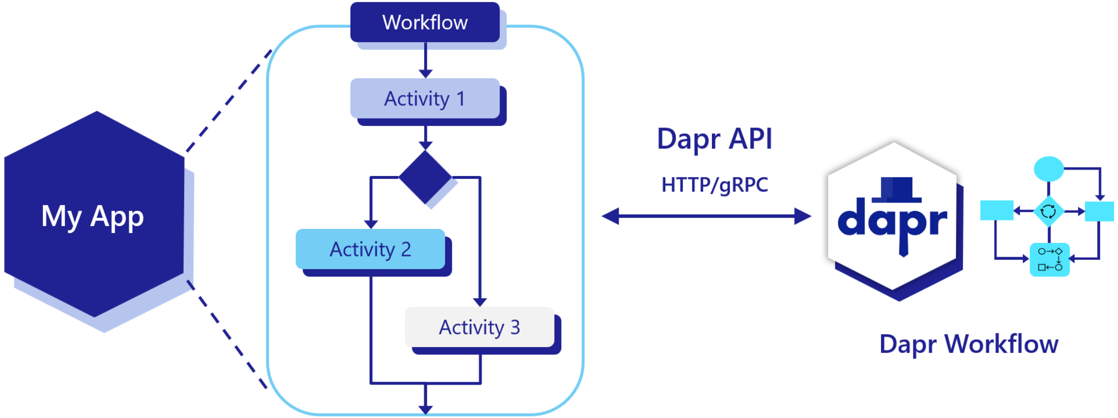


Manual tasks



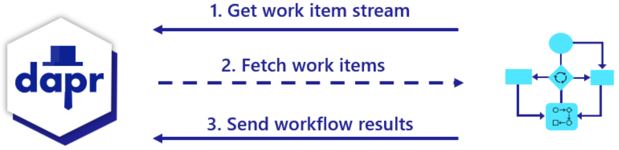
Dapr Workflow architecture

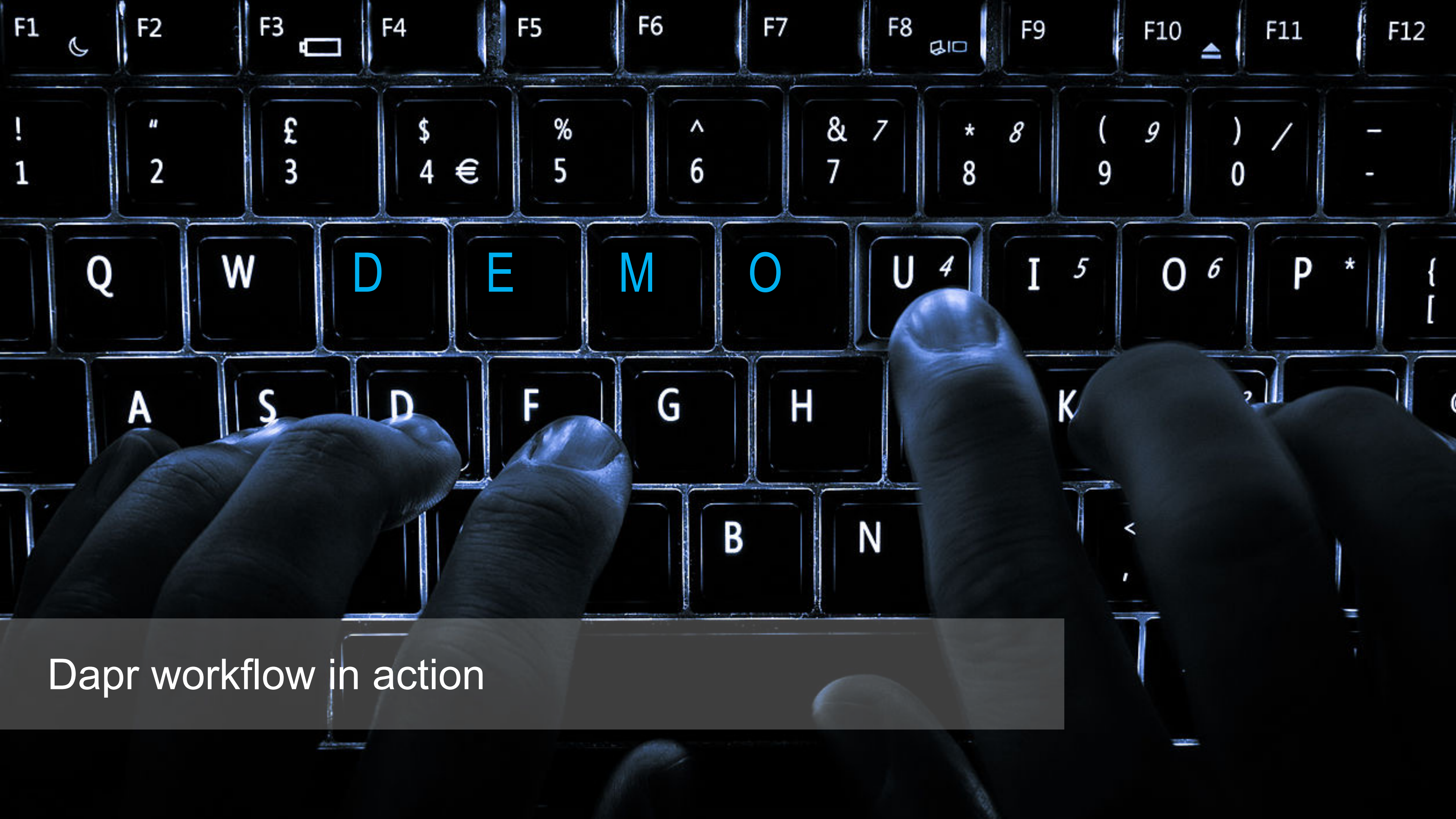
- The workflow engine runs inside of the Dapr sidecar and orchestrates workflow code that is part of your application



Dapr Workflow architecture

- Upon startup, your application connects to the workflow runtime in the Dapr sidecar over a gRPC streaming connection
- It registers the workflow and workflow activities with the workflow engine
- The workflow engine schedules work items that are executed by your application code
  - Application pulls work items from the sidecar so no inbound ports need to opened

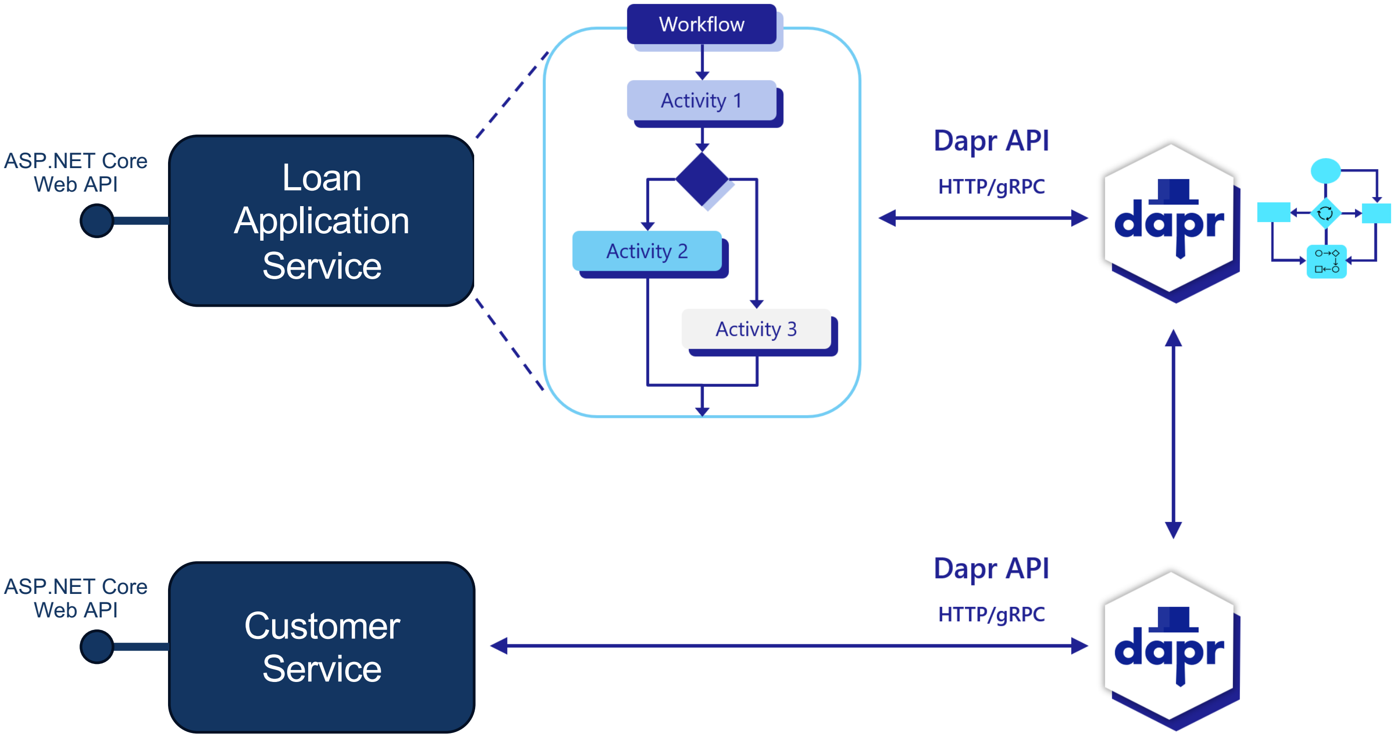




D E M O

Dapr workflow in action

Demo setup


ASP.NET Core
Web API
Loan Application Service
Workflow
Activity 1
Activity 2
Activity 3
Dapr API
HTTP/gRPC
dapr
ASP.NET Core
Web API
Customer Service
Dapr API
HTTP/gRPC
dapr

Thank you!

Edwin van Wijk

Principal Solution Architect / Managing consultant

infoSupport Solid Innovator

MVP Microsoft® Most Valuable Professional

Resources:

Demo code:

https://github.com/edwinvw/dapr-workflow-demos

All things Dapr:

https://dapr.io
https://docs.dapr.io

Free e-book**: Dapr for .NET Developers**

https://aka.ms/dapr-ebook

Microsoft

Dapr for
.NET Developers

Foreword by Mark Russinovich, Microsoft Azure CTO and Technical Fellow

dapr

Robert Vettor
Sander Molenkamp
Edwin van Wijk

eShopOnDapr reference case:

https://github.com/dotnet-architecture/eShopOnDapr

Dapr Traffic Control sample app:

https://github.com/edwinvw/dapr-traffic-control