Azure Durable Functions Lessons learned

Vladimir Horbovanu

CTO, Codexcite

Monterrey

.NET

User Group

Vlad Horbovanu - Azure Durable Functions: Lecciones aprendidas

Marzo 18, 2021

Monterrey
.NET
User Group

Vladimir Horbovanu

- Romanian
- 17 years in Monterrey
- 20+ years experience as a developer
- 10+ years experience as an entrepreneur
- Passionate about code
- CTO - Codexcite

Agenda

- Azure Functions overview
- Durable Functions introduction
- Implementation insights from real-life development

Azure Functions overview

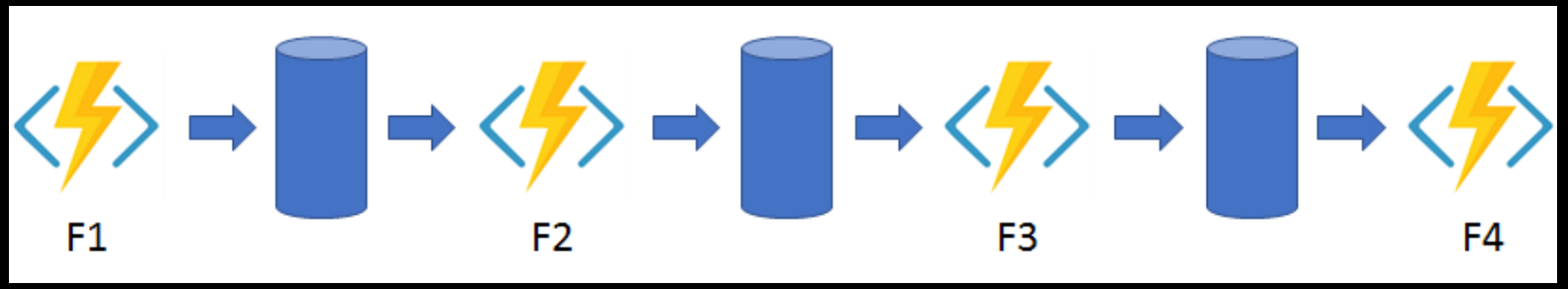
- Event driven
- Compute-on-demand
- Serverless
- Functions must not have state
  - So they can scale as needed
- Functions must be short lived
  - You pay for the time they are running

Durable Functions introduction

- Write stateful functions in a serverless compute environment
- The extension manages state, checkpoints, and restarts
- You can focus on business logic

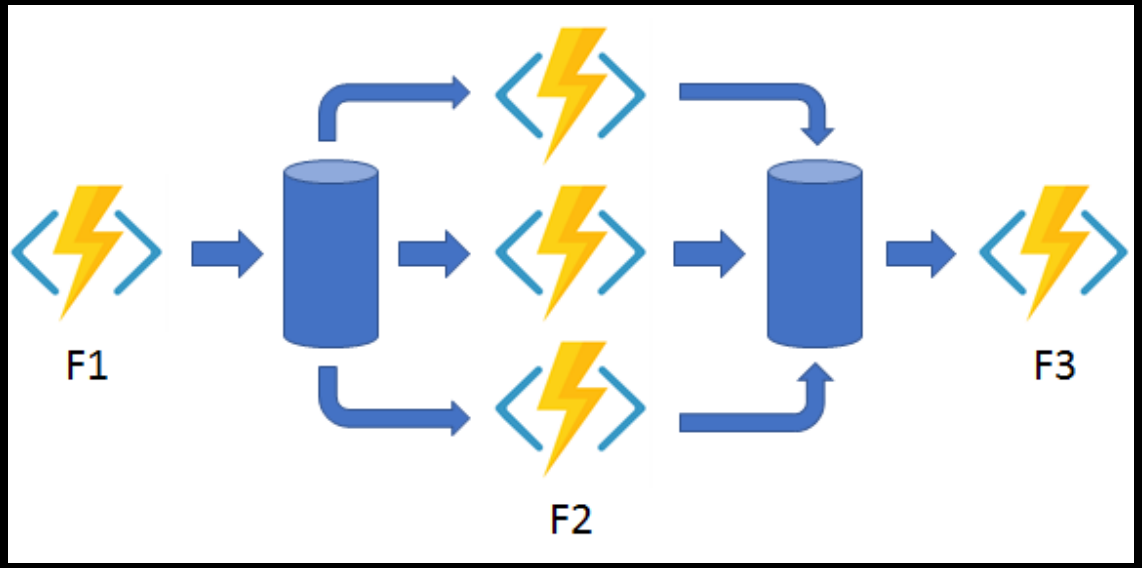
Common patterns

Function chaining



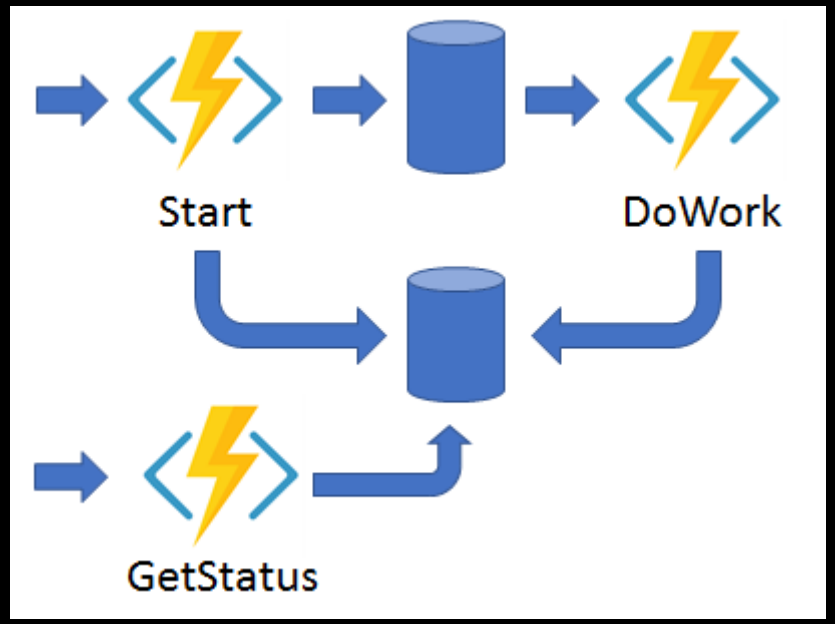
Common patterns

Fan out / fan in



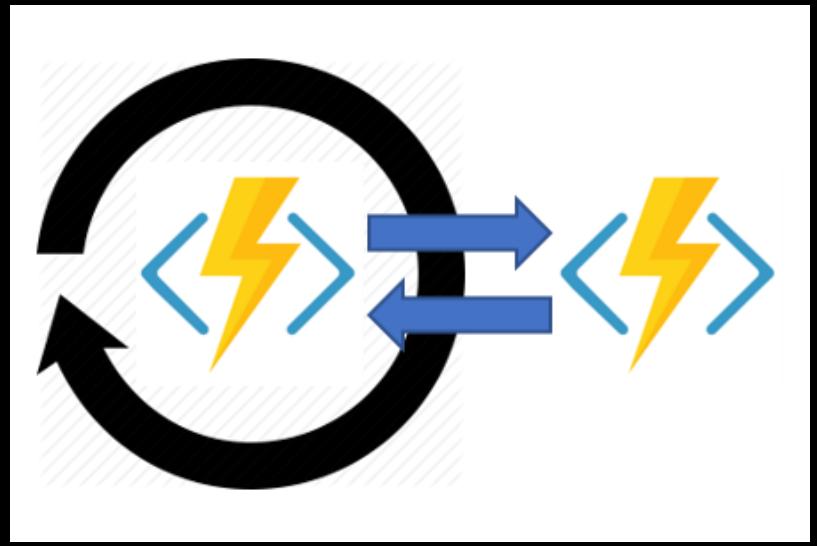
Common patterns

Async HTTP APIs



Common patterns

Monitor

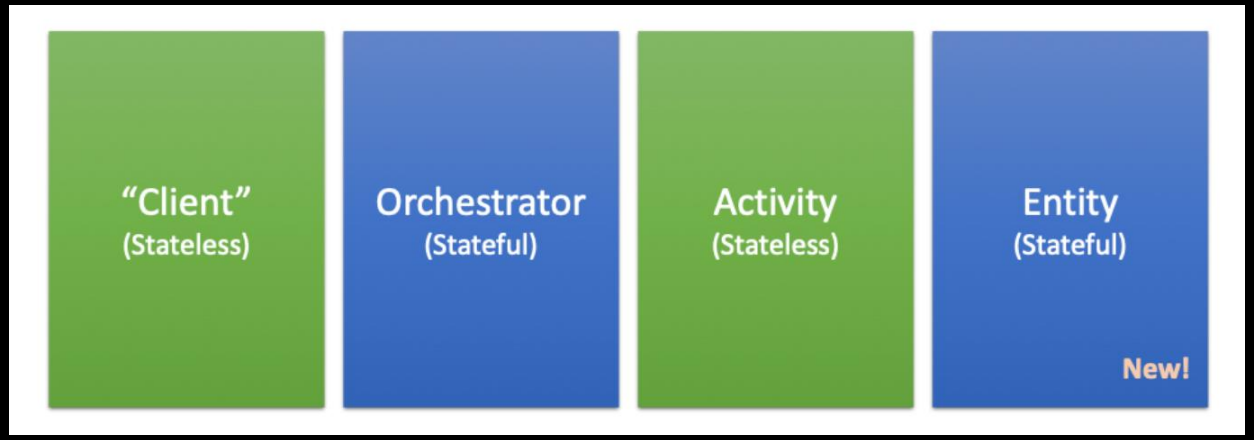


Common patterns

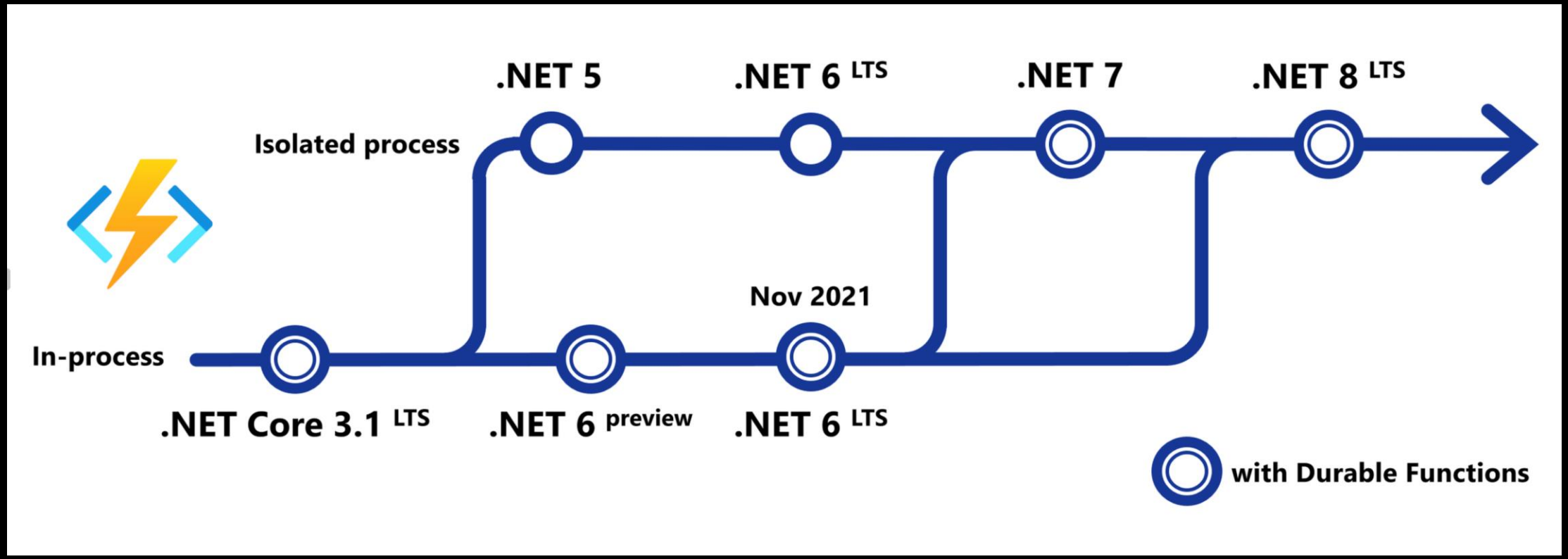
Human interaction

Durable Functions types

- Client function
  - Can use any other function trigger
  - Starts or signals an orchestrator
- Orchestrator function
  - Defines the workflow
  - Can call activities, sub-orchestrations, entities
- Activity function
  - Can run async operations
  - Called by orchestrators
- Entity functions
  - Actor-like capabilities
  - One-at-a-time calls per entity



Roadmap


.NET 5
.NET 6 LTS
.NET 7
.NET 8 LTS
Isolated process
Nov 2021
In-process
.NET Core 3.1 LTS
.NET 6 preview
.NET 6 LTS
with Durable Functions

Implementation insights

- Project requirements
  - User prequalifies online
    - Can take some time
    - Dependencies can fail or be offline
    - Retry mechanism required
  - User approval needed to proceed to agent assignment
    - Several simultaneous options for the approval
      - Web app click
      - Whatsapp message
  - Agents bid on the lead to win the assignment
    - Via Whatsapp

Simple Orchestration

- async in workflows
  - calling async methods it's ok
  - if they respect the same rules
- make workflows easy to read
- auth level - don't forget when publishing
- Logging
  - save your instanceId, isReply and parent instanceId in the log details
  - isReply is important!
- consider the overhead of calling activities

Custom Status Orchestration

- DateTime
  - all in UTC
  - but not DateTimeOffset
- waiting for determinate time
  - it's about waiting until a specific time
- custom status
  - useful for ongoing updates

Parameter Orchestration

- passing parameters
  - actually serialize / deserialize
  - be aware of serialization problems
    - datetime, numbers, enums
    - Interfaces, derived classes
  - tuples ok, EXCEPT for start orchestration (workarond)
  - no type checking - but some analyzers
  - id vs whole object
  - Exceptions get serialized too
    - Must have constructor with no parameters

Entity Orchestration

- Entities
  - Works as a locking mechanism
  - You can only call methods with return value from workflows.
  - The other functions can
    - read state
    - send signals (no return value)
- Read-update pattern using entities
  - Avoid concurrency problems
- Throttled activity pattern

Versioned Orchestration

- Changing an orchestrator can lead to errors for in-flight running instances
- Orchestrator versioning pattern

User action Orchestration

- Retry mechanism
  - activity MUST fail – not available for return values
- Try catch
  - Exceptions are wrapped for activity or sub-orchestration fails
- Wait for external event, like user interaction
  - How do you get the instanceId to the user?
- What happens when you can't get the instanceId to the user?
  - Subscriber Entity pattern
- Recheck conditions, remember time might have passed between activity calls

Main takeaway

- Durable functions are awesome!
- Advantage:
  - code like you are in a normal, stateful method
- Disadvantage:
  - you forget you are not REALLY in a stateful method

Questions?

https://github.com/Codexcite/Durable-Functions-Lessons-Learned

Thank you!

Vladimir Horbovanu
vlad@codexcite.com
+528183663497
linkedin.com/in/vladhorby/