

Proactive Serverless Function Resource Management

Sixth International Workshop on Serverless
Computing (WoSC6) 2020

Erika Hunhoff, Shazal Irshad, Vijay Thurimella*, Ali Tariq, Eric Rozner
University of Colorado Boulder, *Thrive, Inc



Background

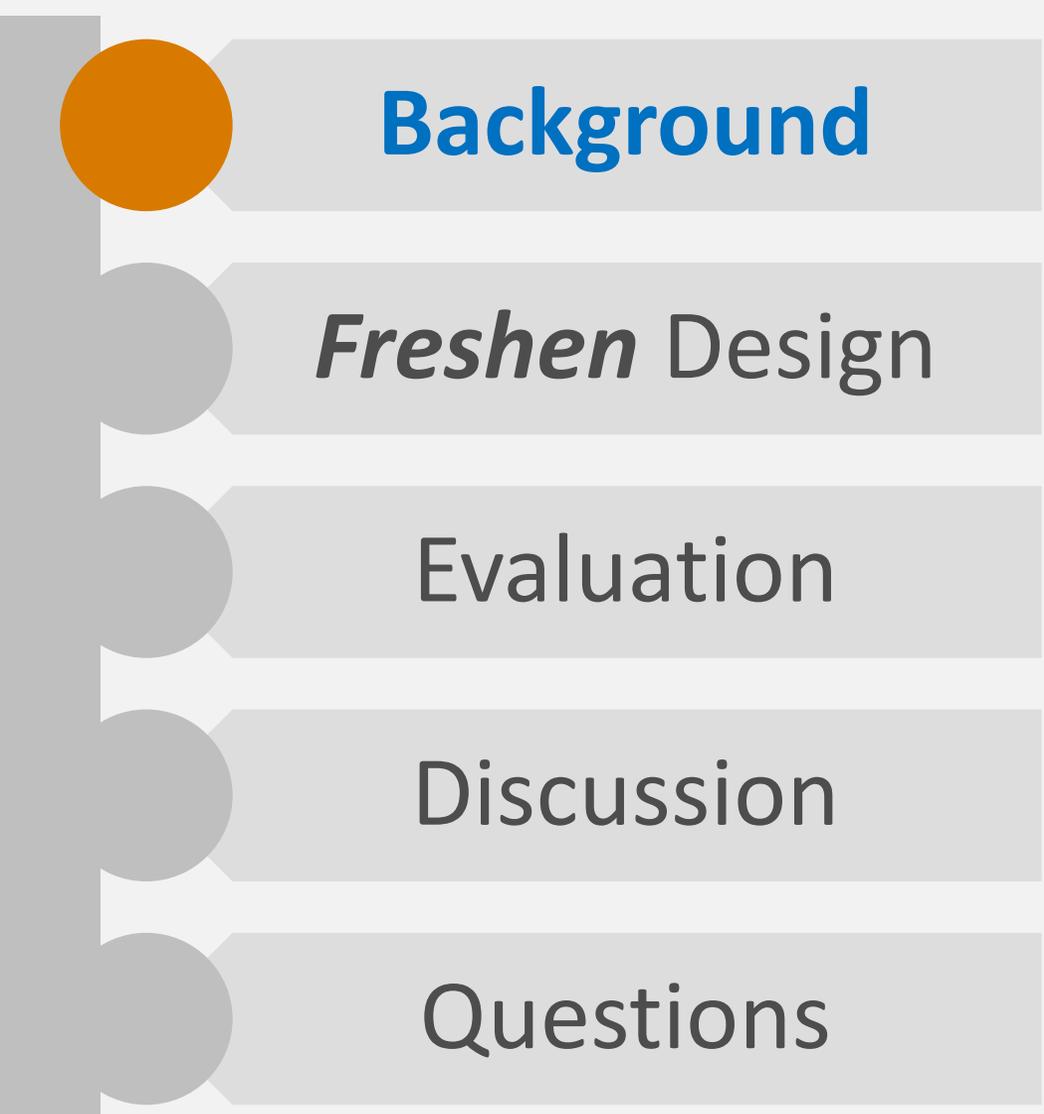
Freshen Design

Evaluation

Discussion

Questions

Outline



Background

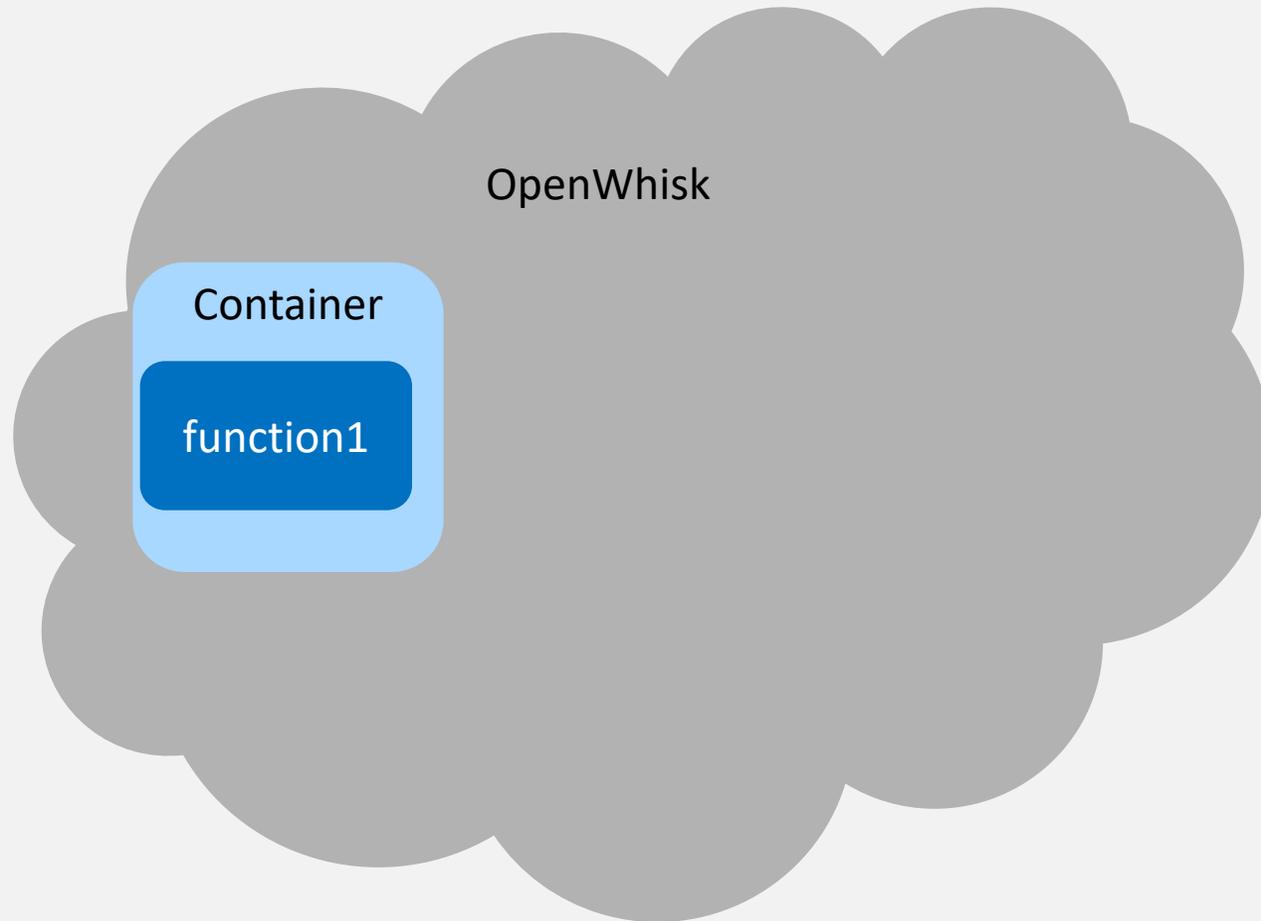
Freshen Design

Evaluation

Discussion

Questions

Outline

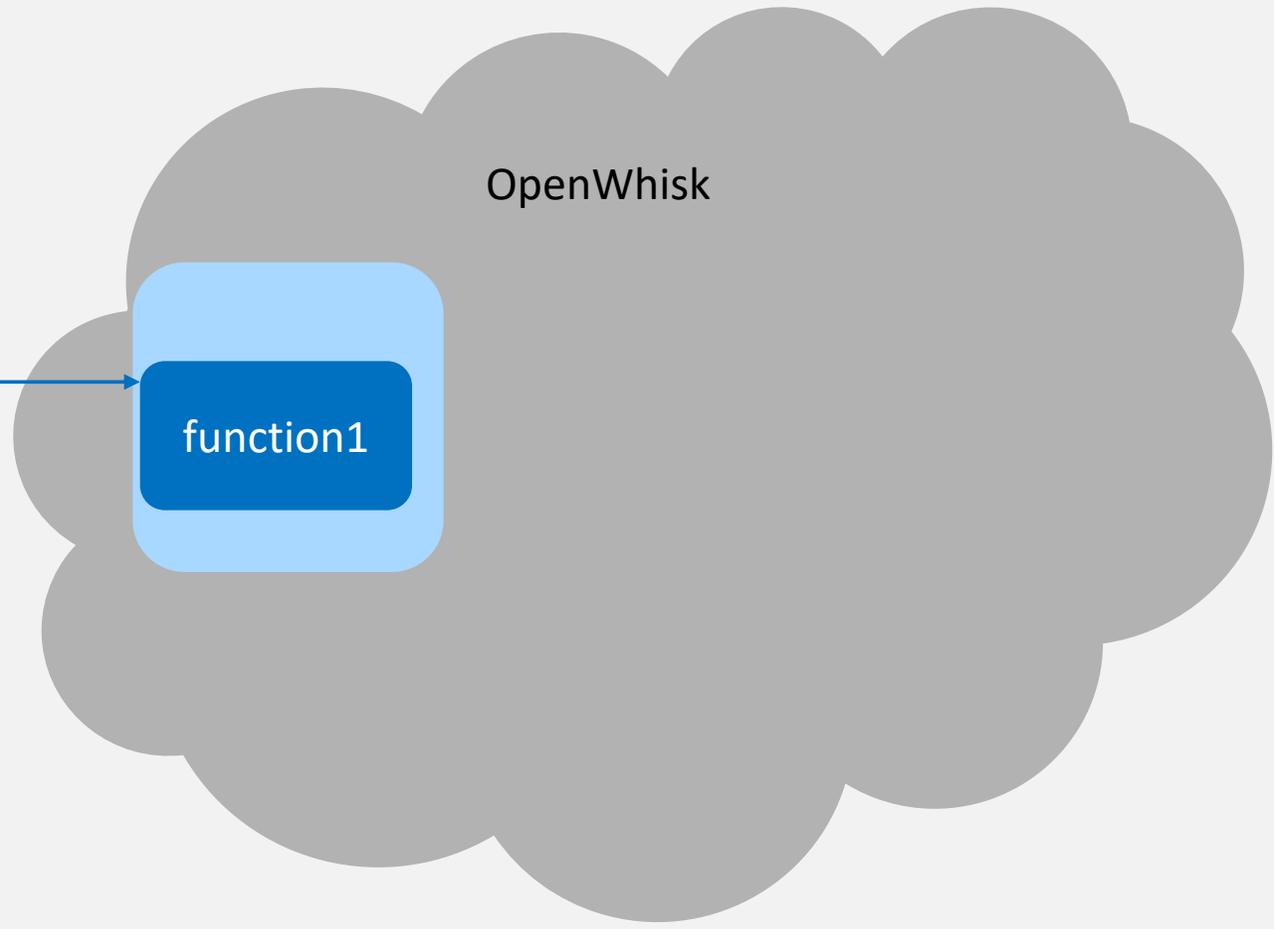


IDCat Serverless Application

UserID: Erika



Function Trigger



OpenWhisk

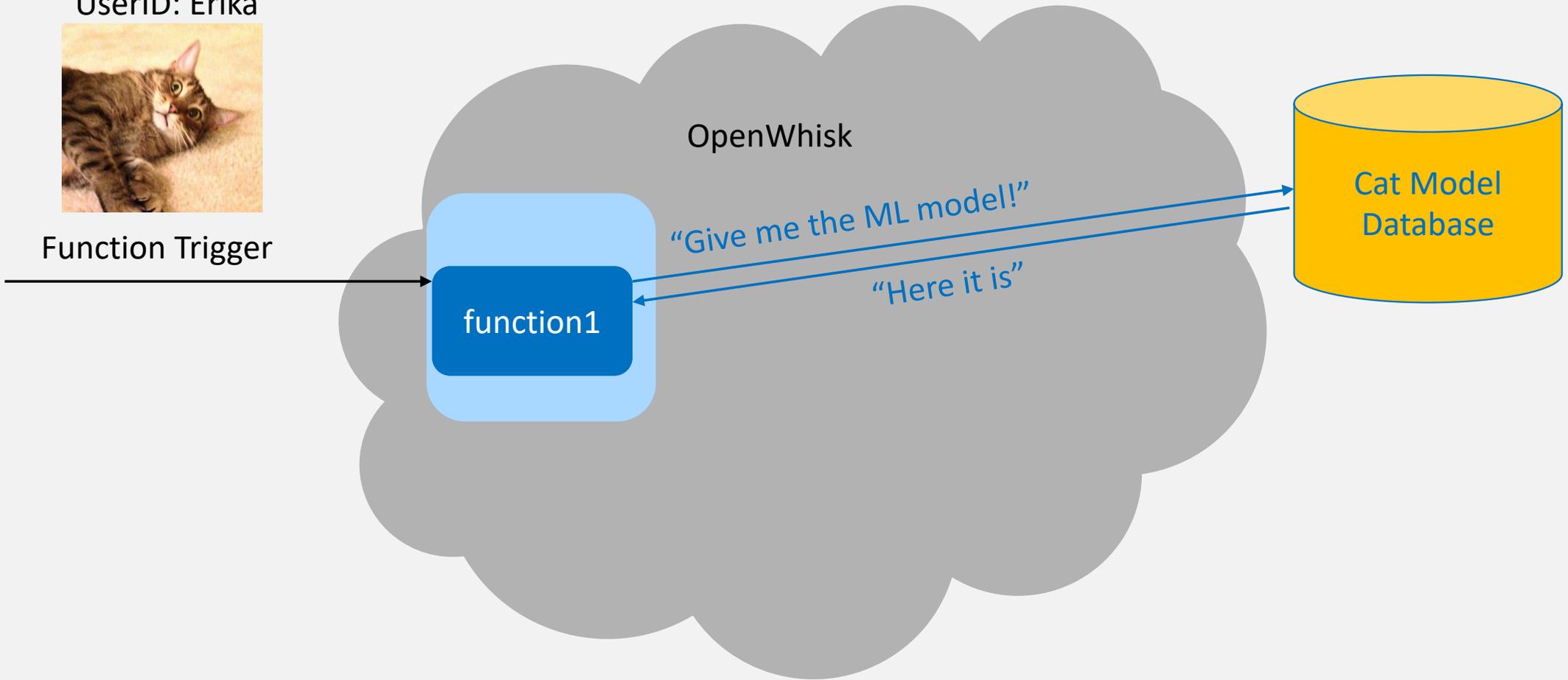
function1

IDCat Serverless Application

UserID: Erika



Function Trigger

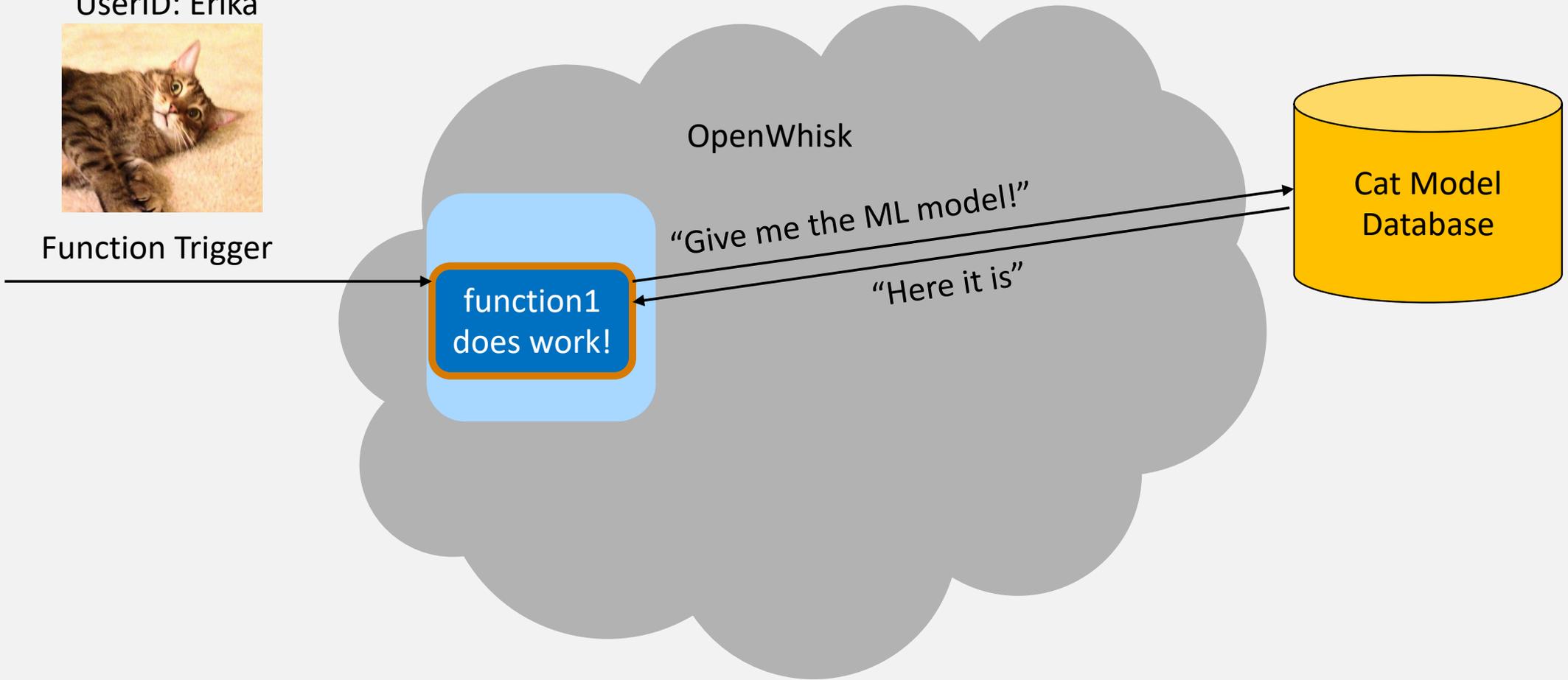


IDCat Serverless Application

UserID: Erika



Function Trigger

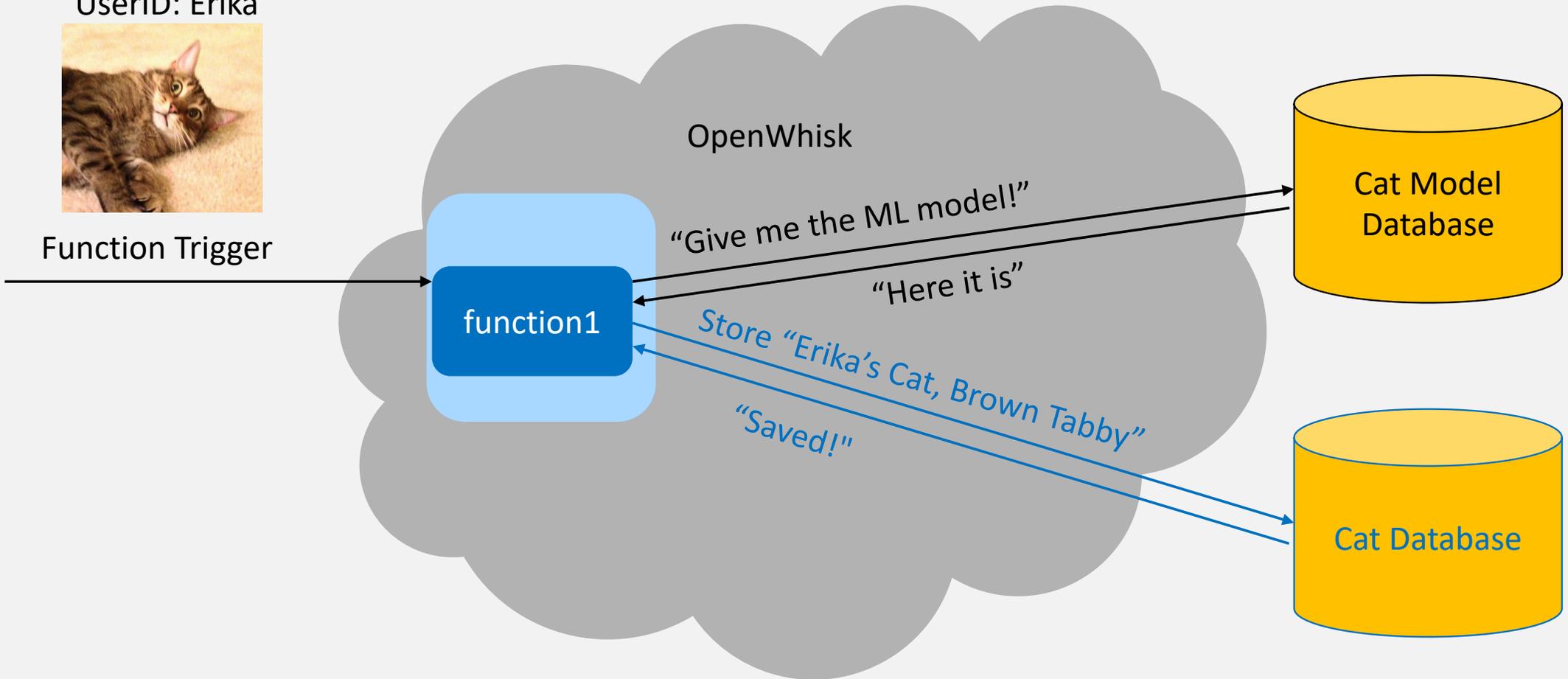


IDCat Serverless Application

UserID: Erika



Function Trigger



IDCat Serverless Application

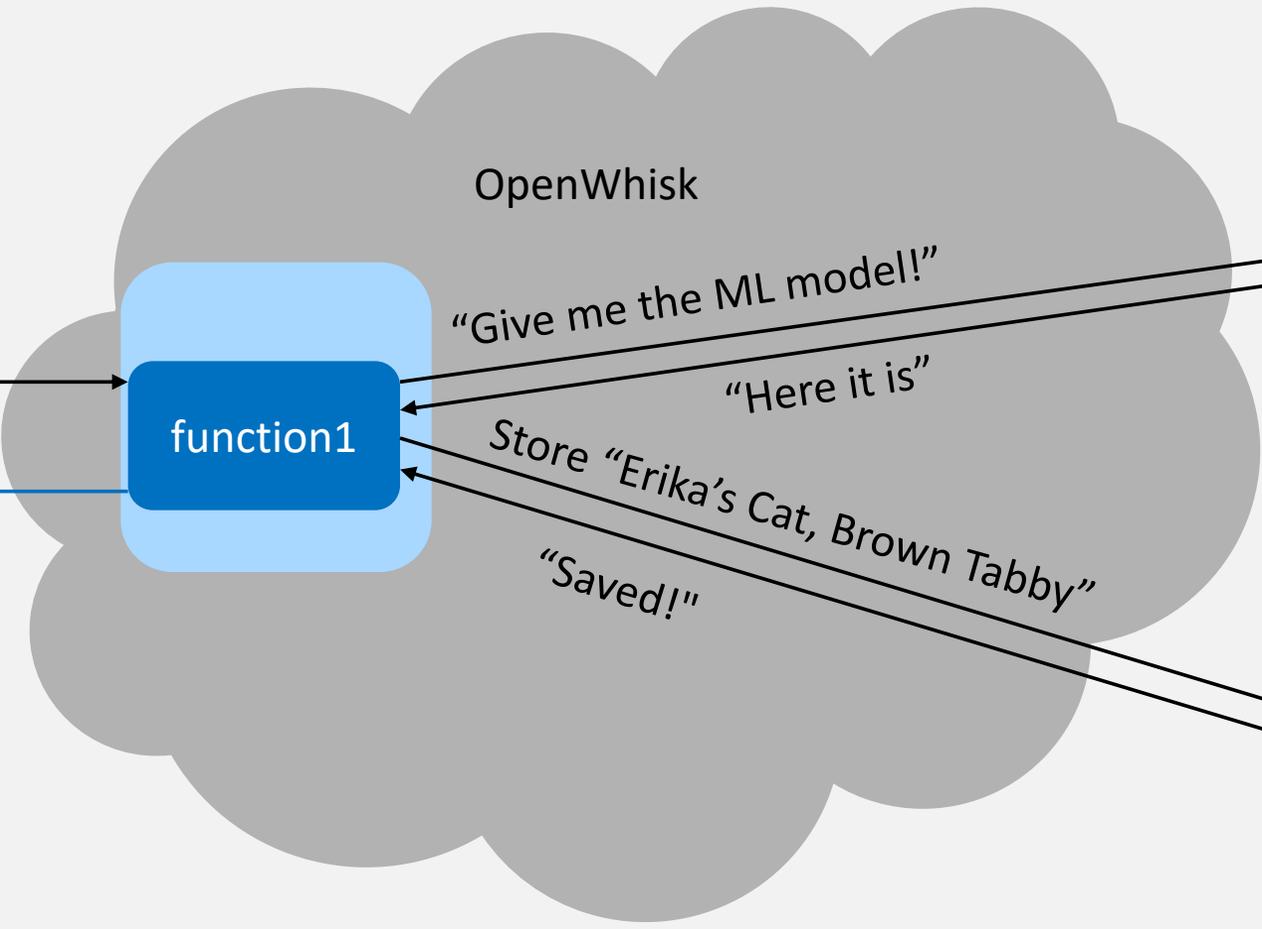
UserID: Erika



Function Trigger



function1



OpenWhisk

"Give me the ML model!"

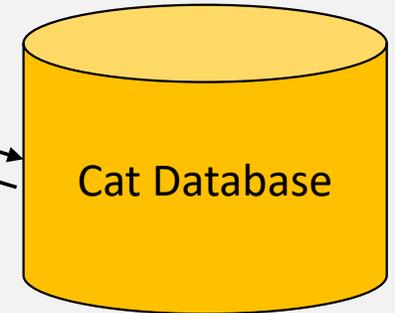
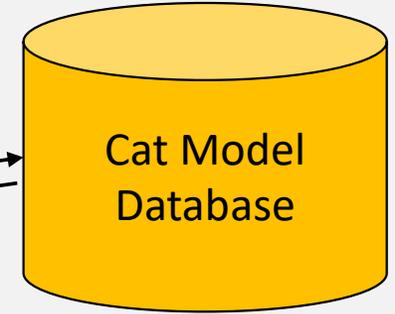
"Here it is"

Store "Erika's Cat, Brown Tabby"

"Saved!"



Result: "Success!"



IDCat Serverless Application

UserID: Erika



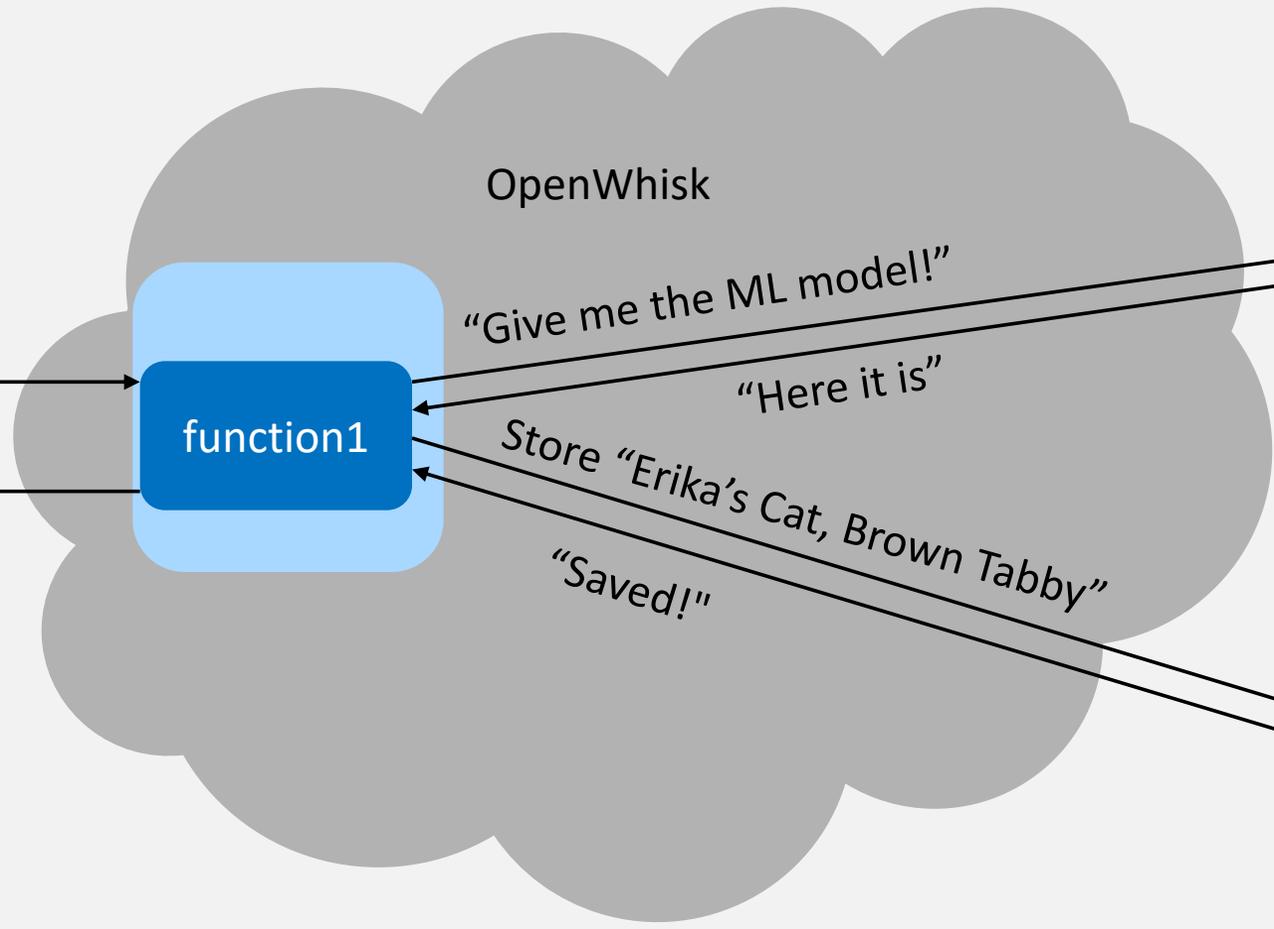
Function Trigger



function1



Result: "Success!"



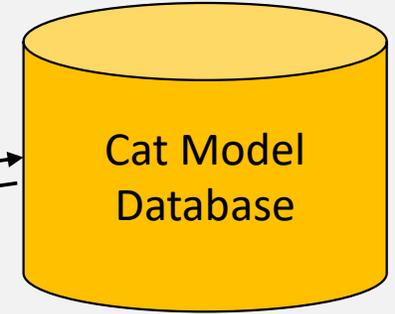
OpenWhisk

"Give me the ML model!"

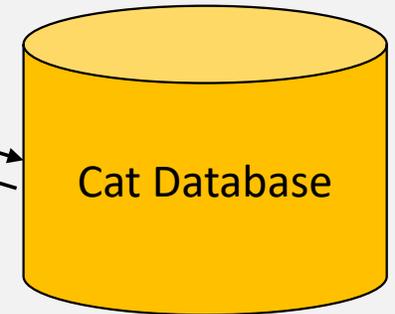
"Here it is"

Store "Erika's Cat, Brown Tabby"

"Saved!"



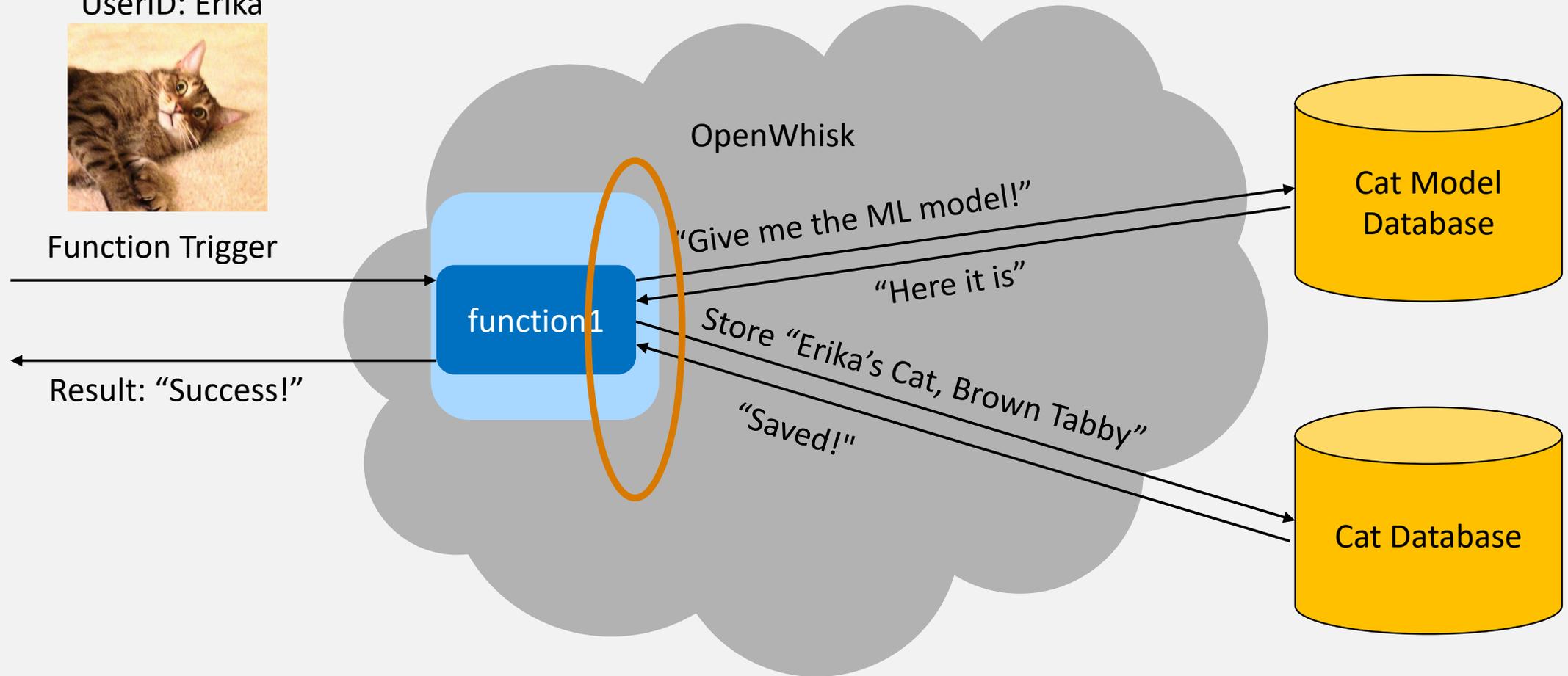
Cat Model Database



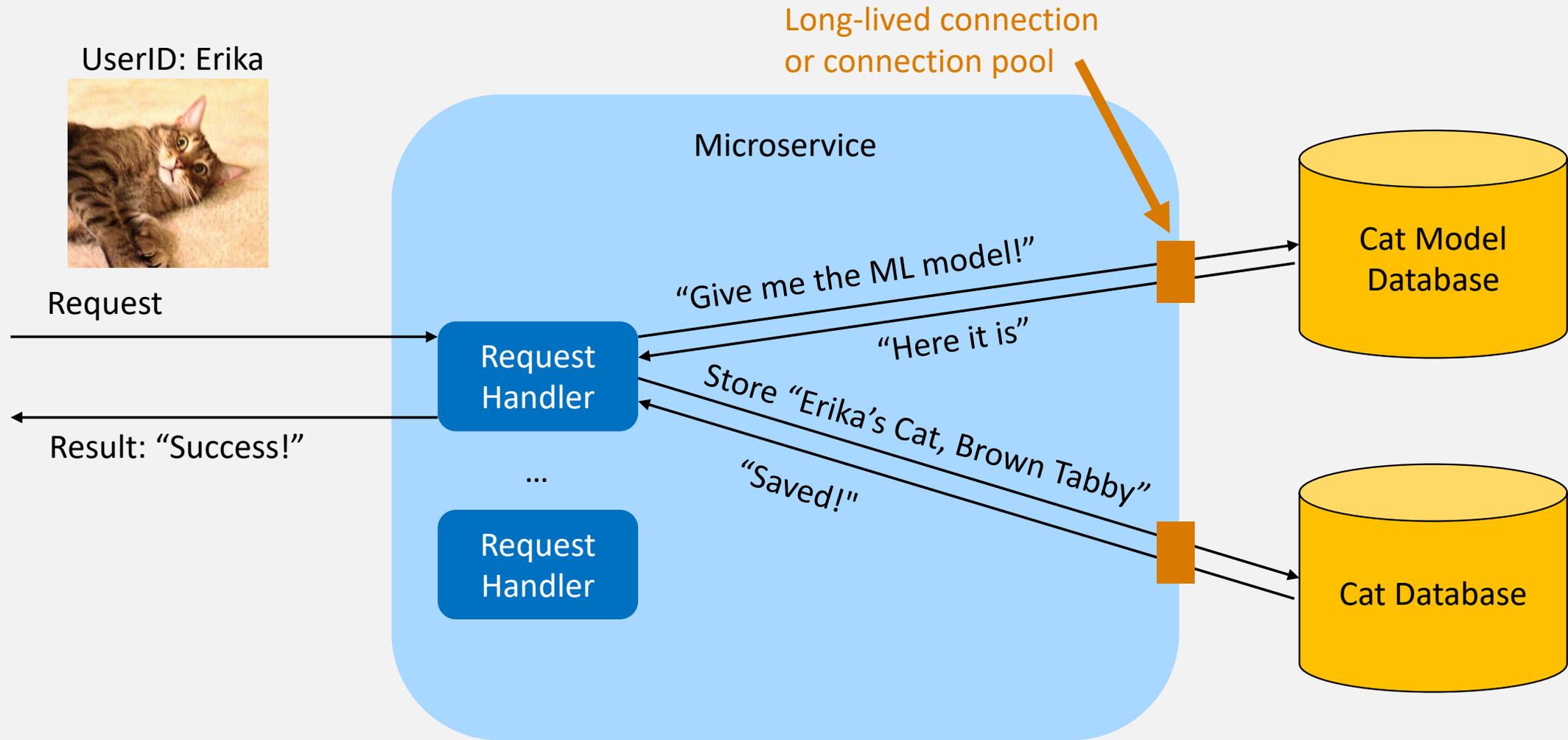
Cat Database

Improve Efficiency?

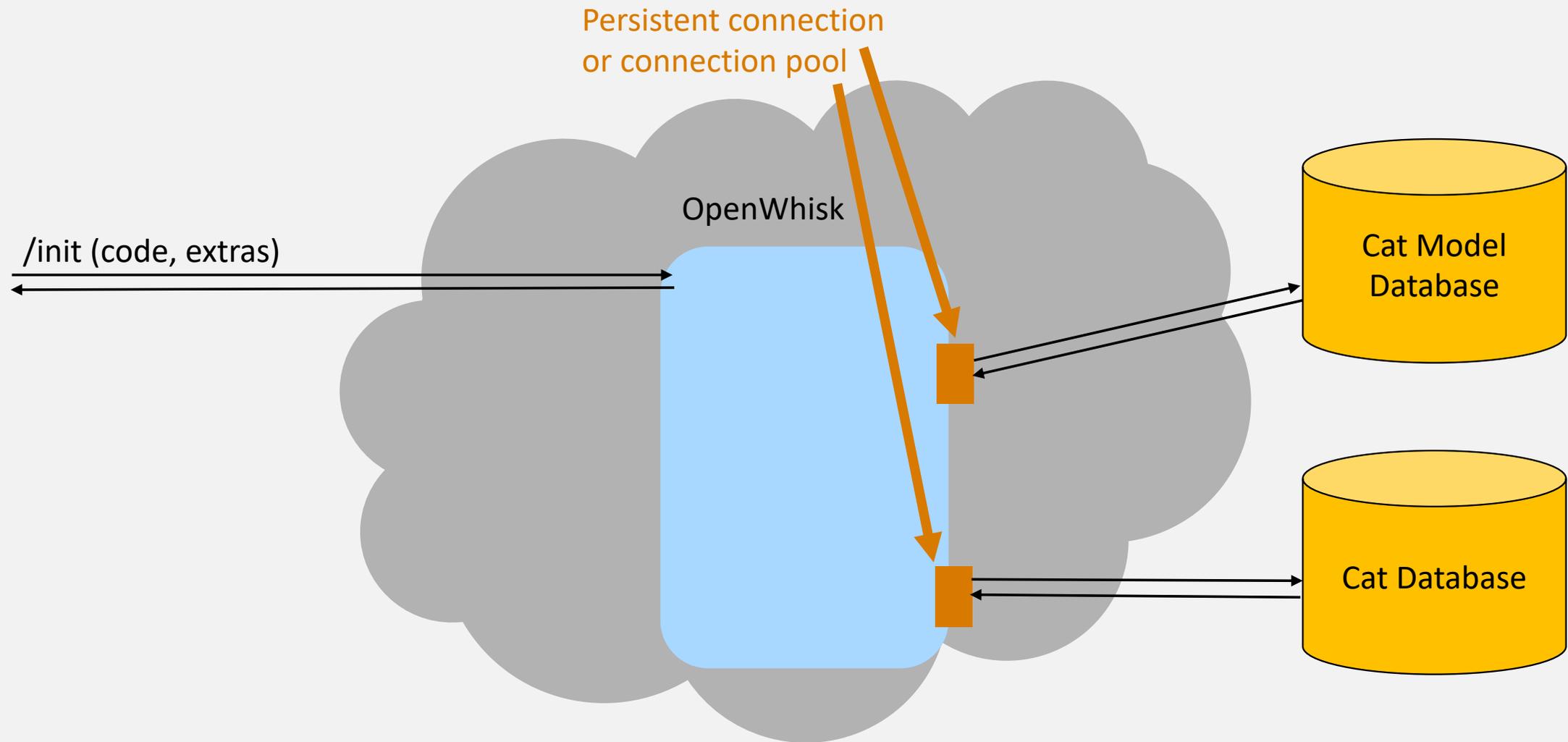
UserID: Erika



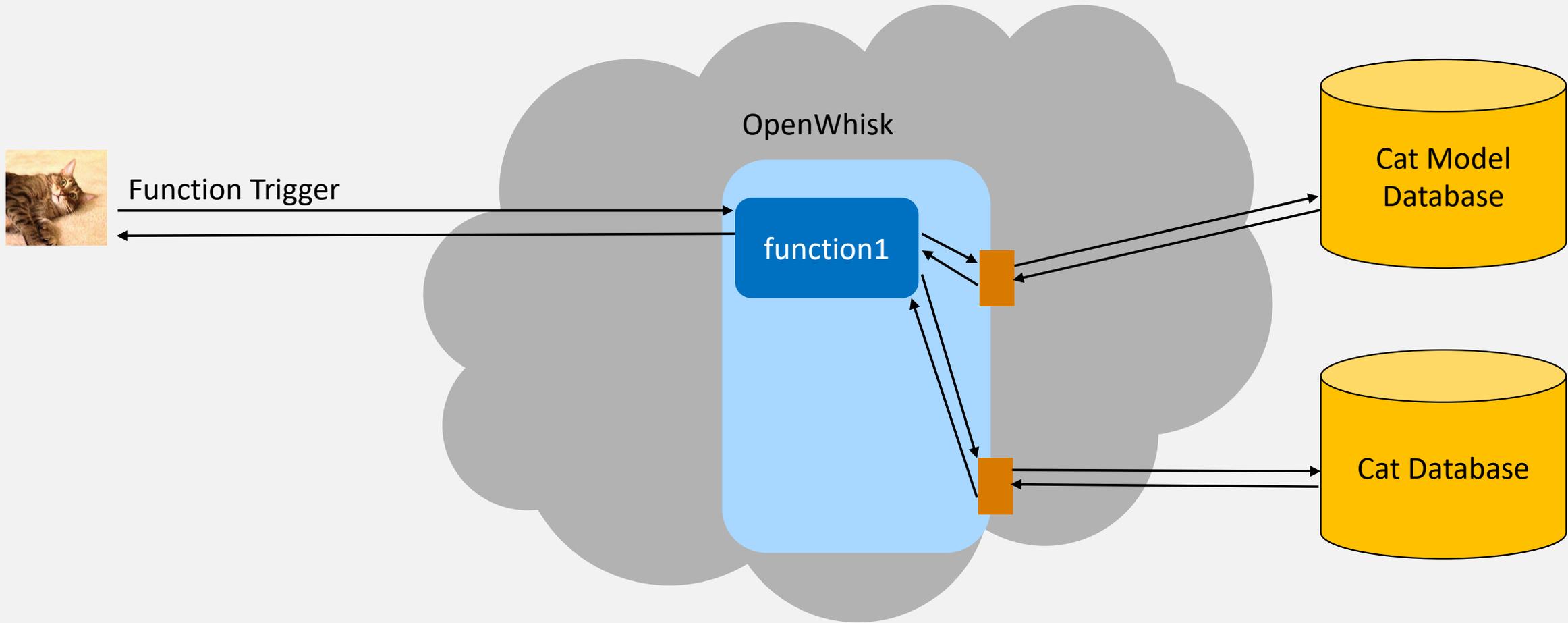
Improve Efficiency?



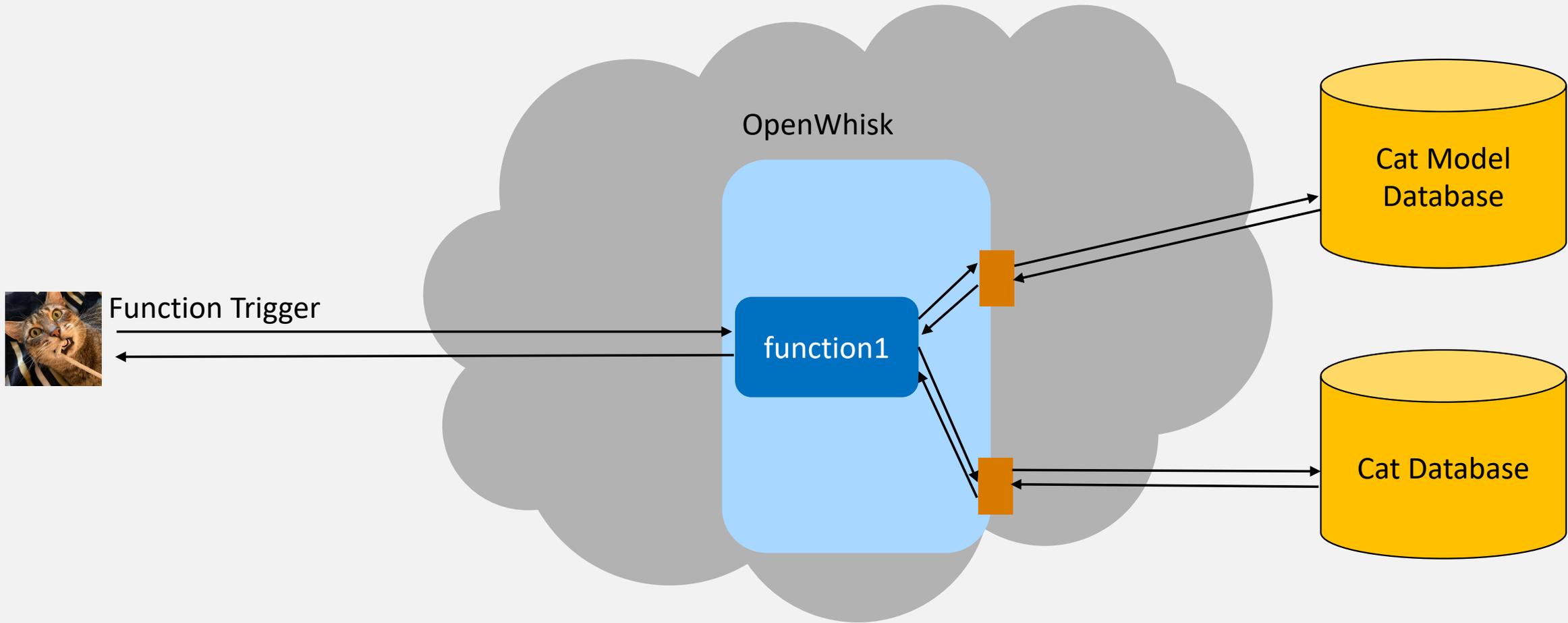
IDCat Microservice



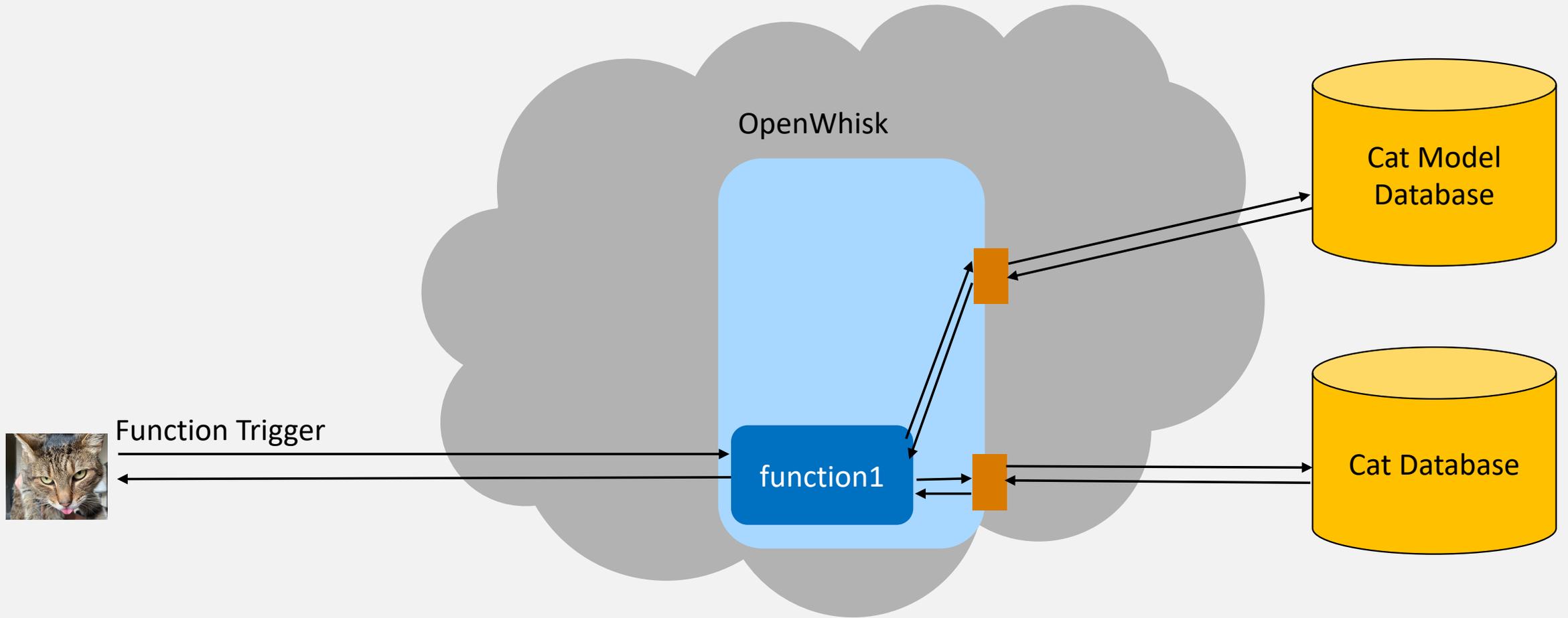
Runtime Reuse in Serverless



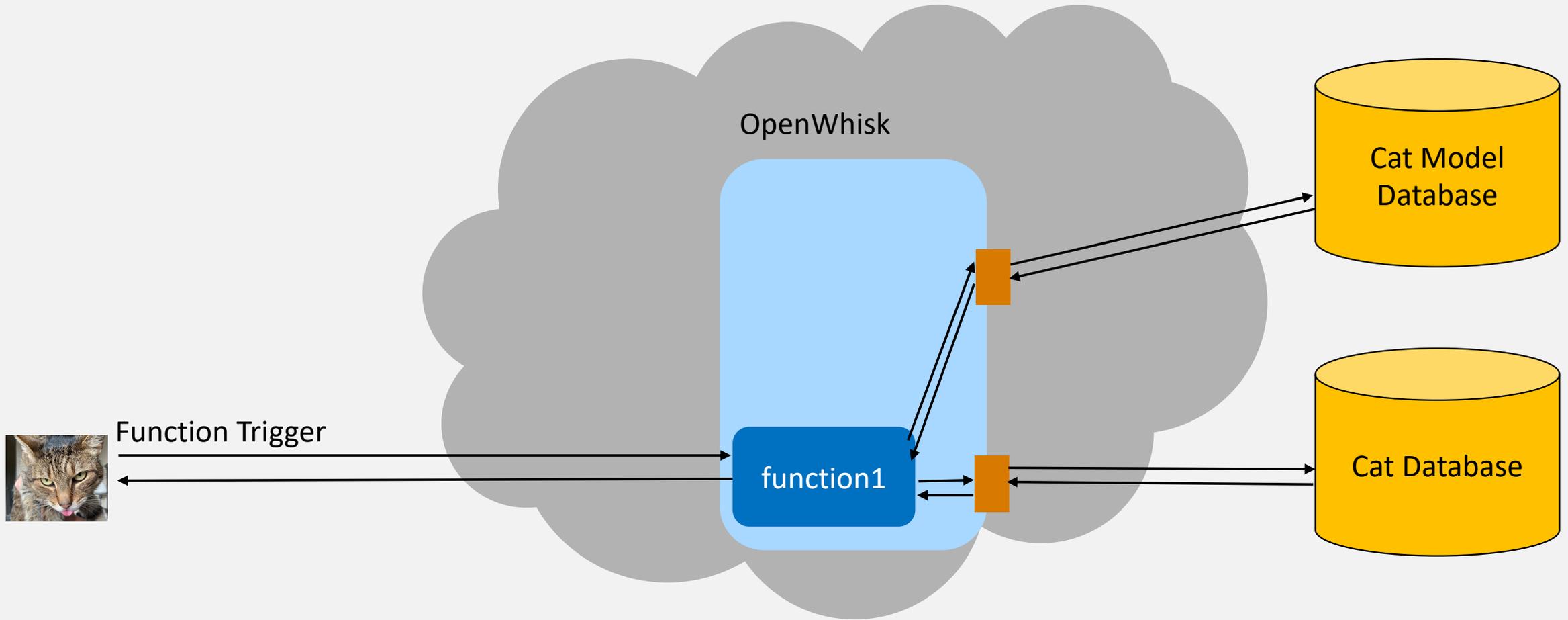
Runtime Reuse in Serverless



Runtime Reuse in Serverless



Runtime Reuse in Serverless



Runtime Reuse in Serverless – **Room for Improvement?**

Background

Freshen Design

Evaluation

Discussion

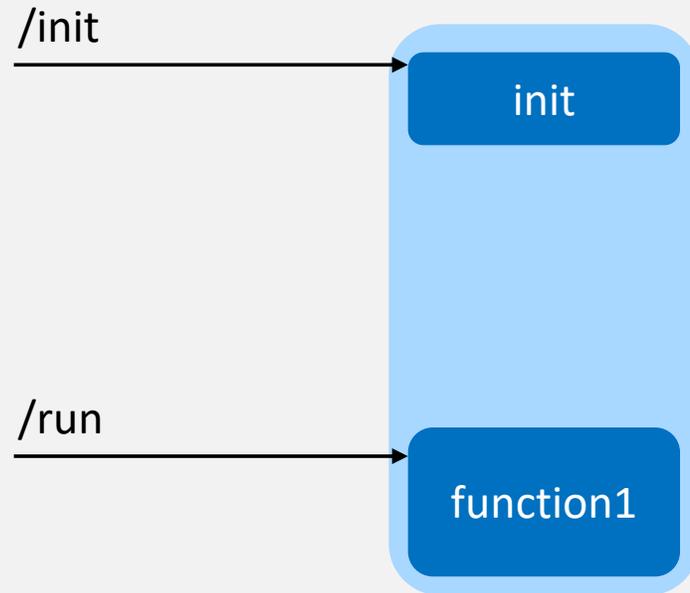
Questions

Outline

- We propose a new serverless runtime primitive, *freshen*, as a mechanism to enable proactive serverless function resource management.

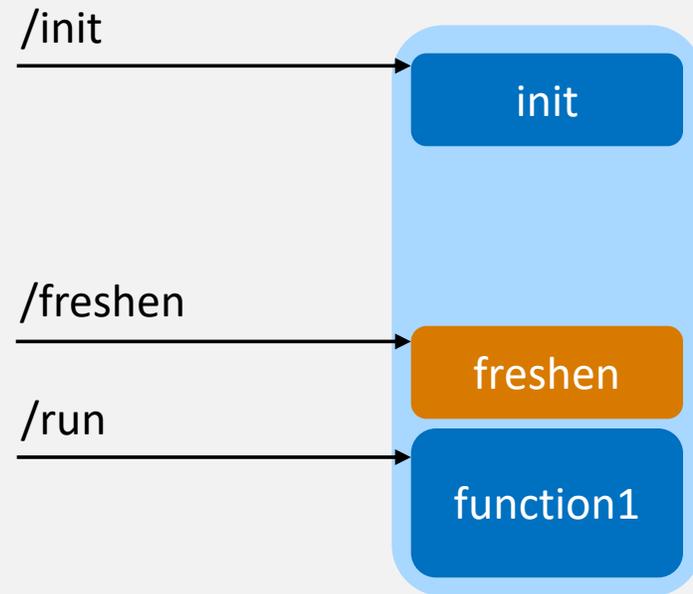
Overview

Time=0



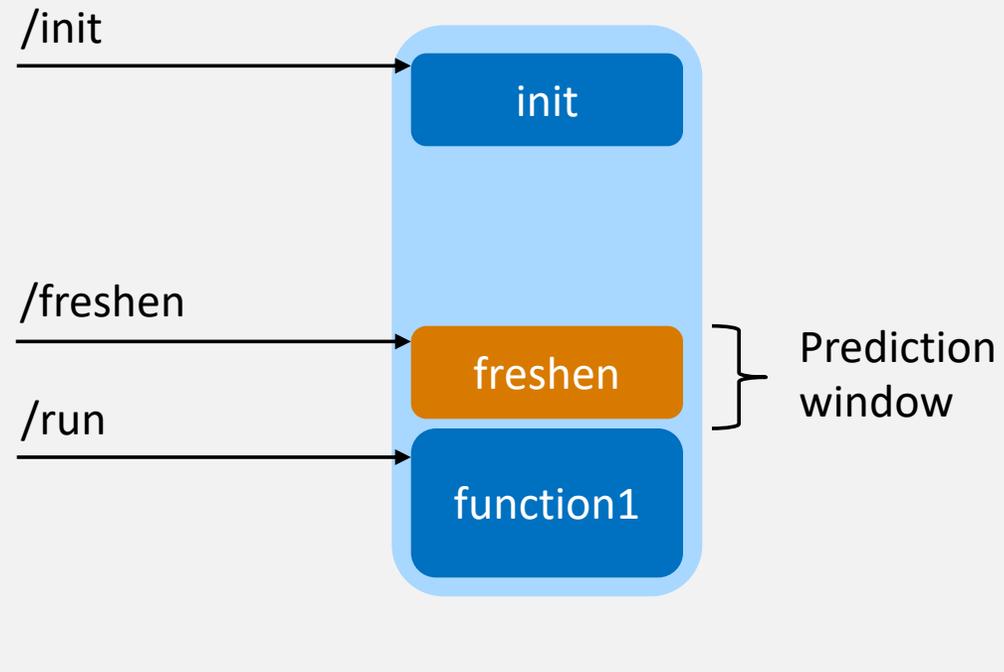
Freshen Design

Time=0



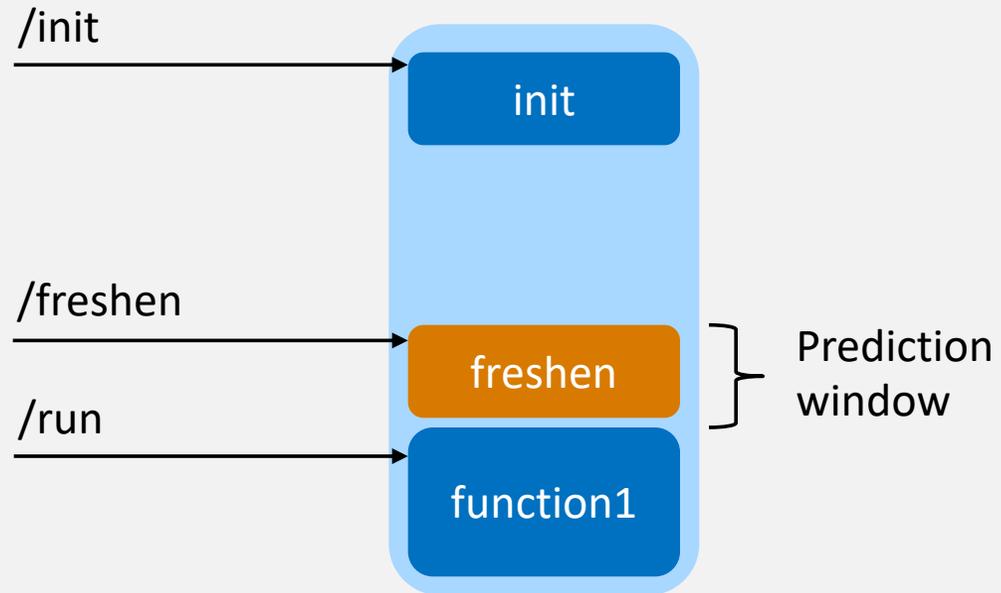
Freshen Design

Time=0



Freshen Design

Time=0

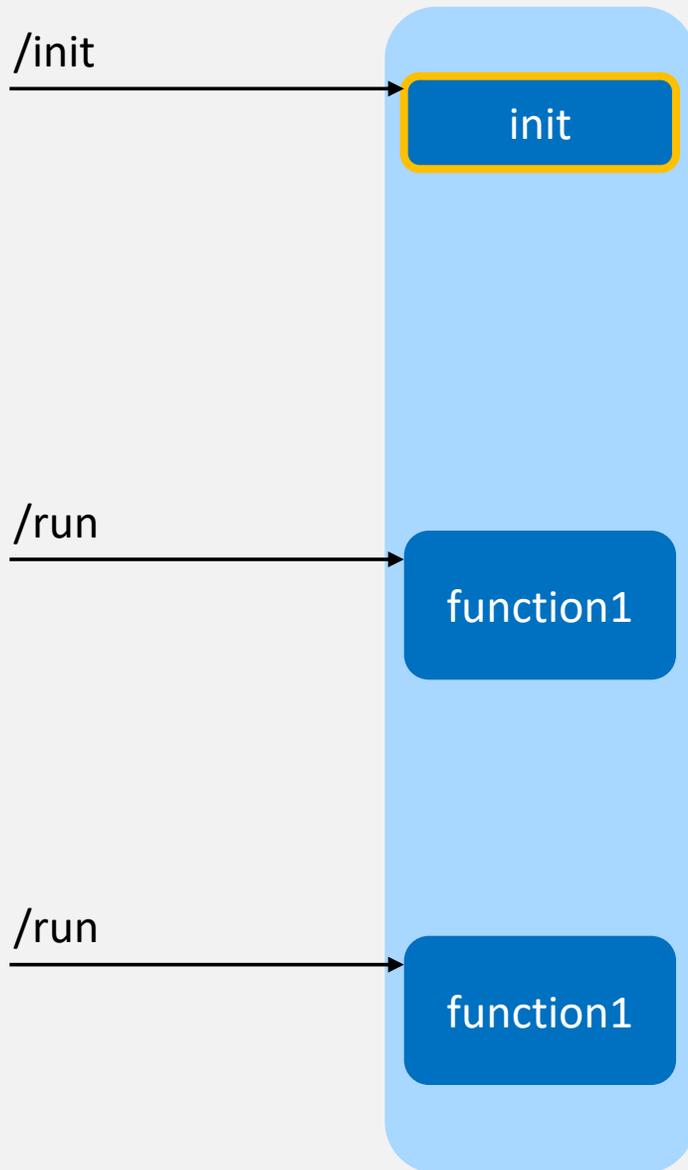


Policy Options:

- Prediction
- Concurrency
- Forced blocking

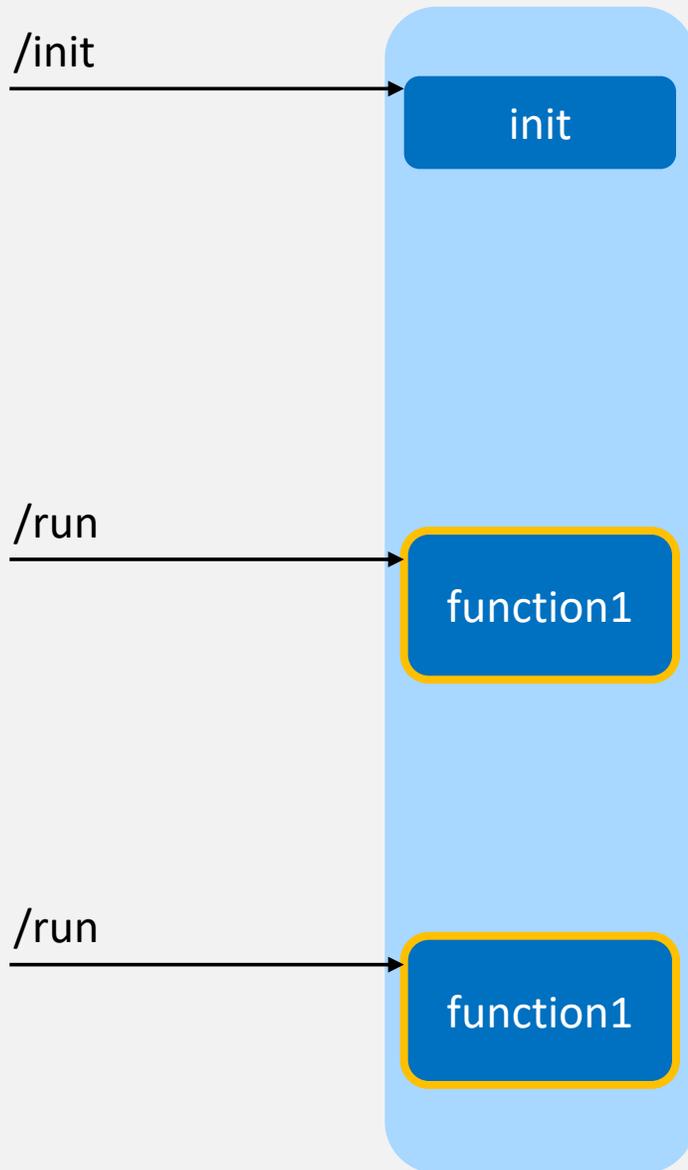
Freshen Design

Time=0



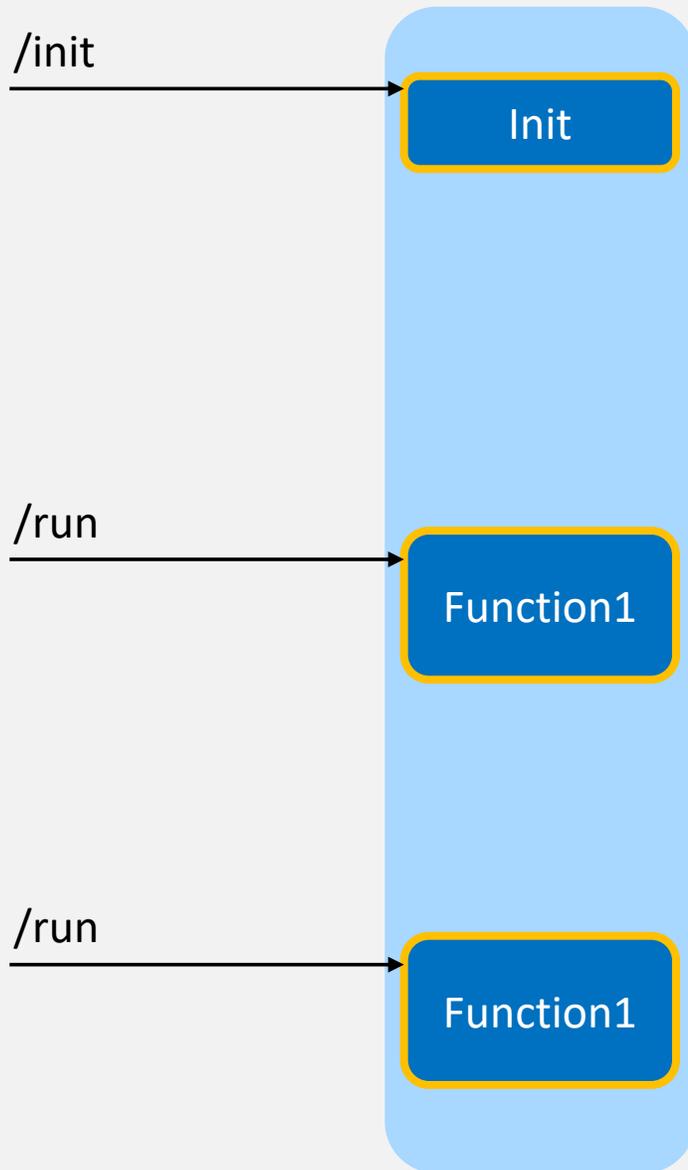
	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code			
Runtime Reuse			
<i>Freshen</i>			

Time=0



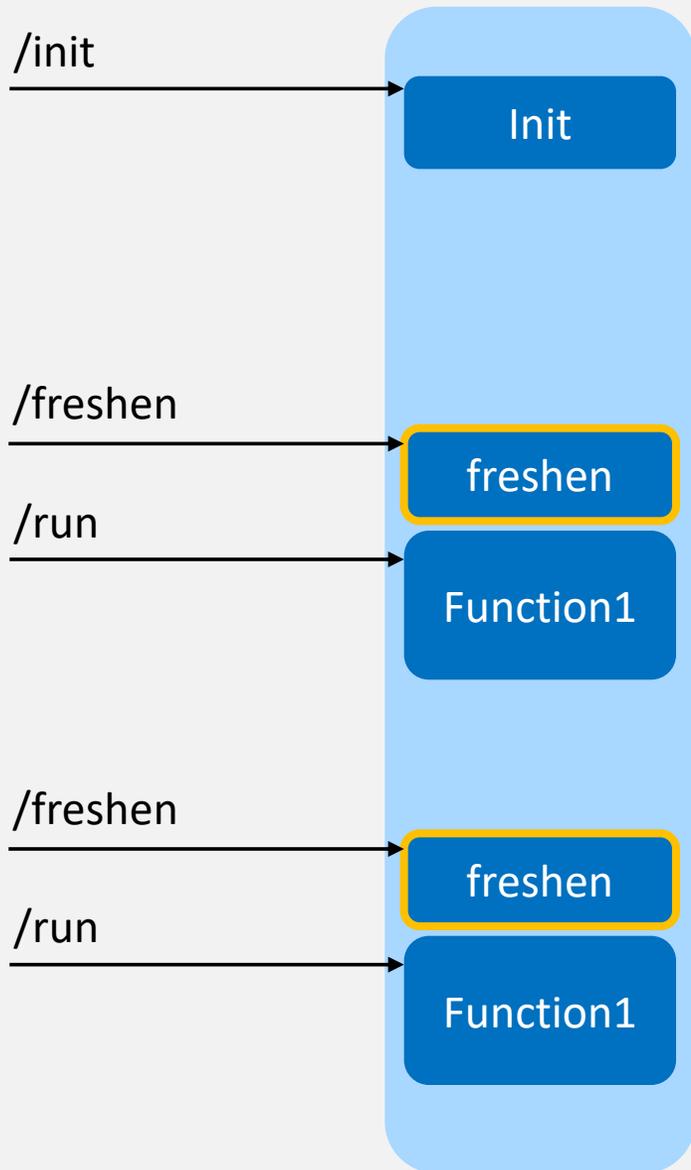
	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code	✗	✓	✗
Runtime Reuse			
<i>Freshen</i>			

Time=0



	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code	✗	✓	✗
Runtime Reuse	✓	✓	✗
<i>Freshen</i>			

Time=0



	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code	✗	✓	✗
Runtime Reuse	✓	✓	✗
<i>Freshen</i>	✓	✓	✓

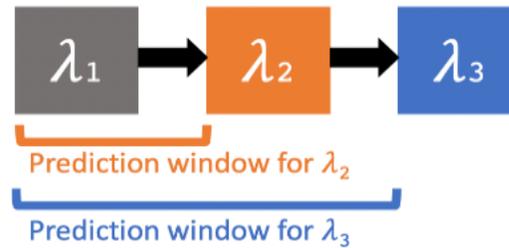
Serverless Function Prediction

Prediction useful for many reasons (scheduling, resource utilization, coldstart avoidance, etc.)

Serverless Function Prediction

Prediction useful for many reasons (scheduling, resource utilization, coldstart avoidance, etc.)

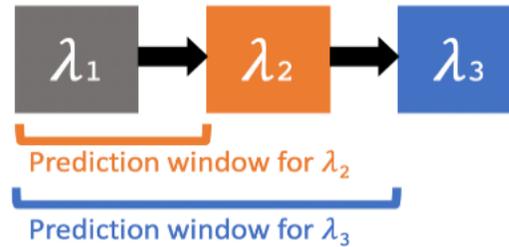
Some cases are easier to predict, e.g., chained functions



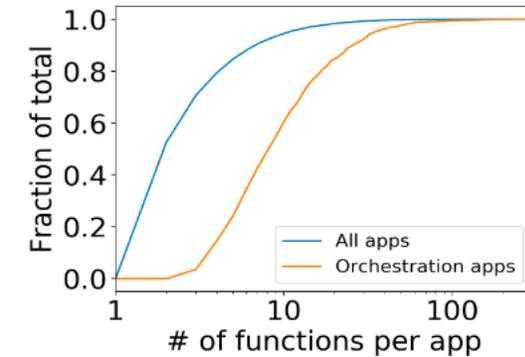
Serverless Function Prediction

Prediction useful for many reasons (scheduling, resource utilization, coldstart avoidance, etc.)

Some cases are easier to predict, e.g., chained functions



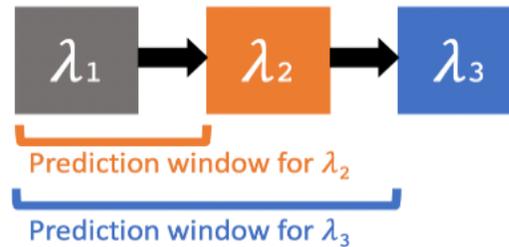
Many applications consist of multiple functions



Serverless Function Prediction

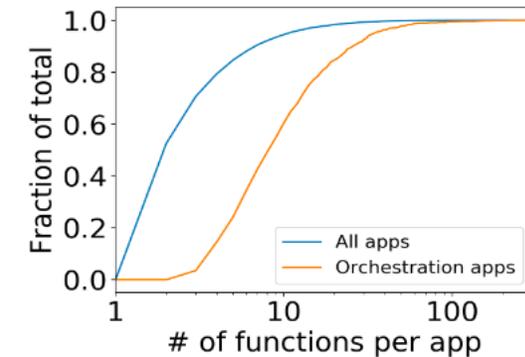
Prediction useful for many reasons (scheduling, resource utilization, coldstart avoidance, etc.)

Some cases are easier to predict, e.g., chained functions



May be
infrastructure
overheads

Many applications consist of multiple functions



Trigger Service	Delay (s)
Step Functions	0.064
Direct (Boto3)	0.060
SNS Pub/Sub	0.253
S3 bucket	1.282

What Can *Freshen* Do?

UserID: Erika



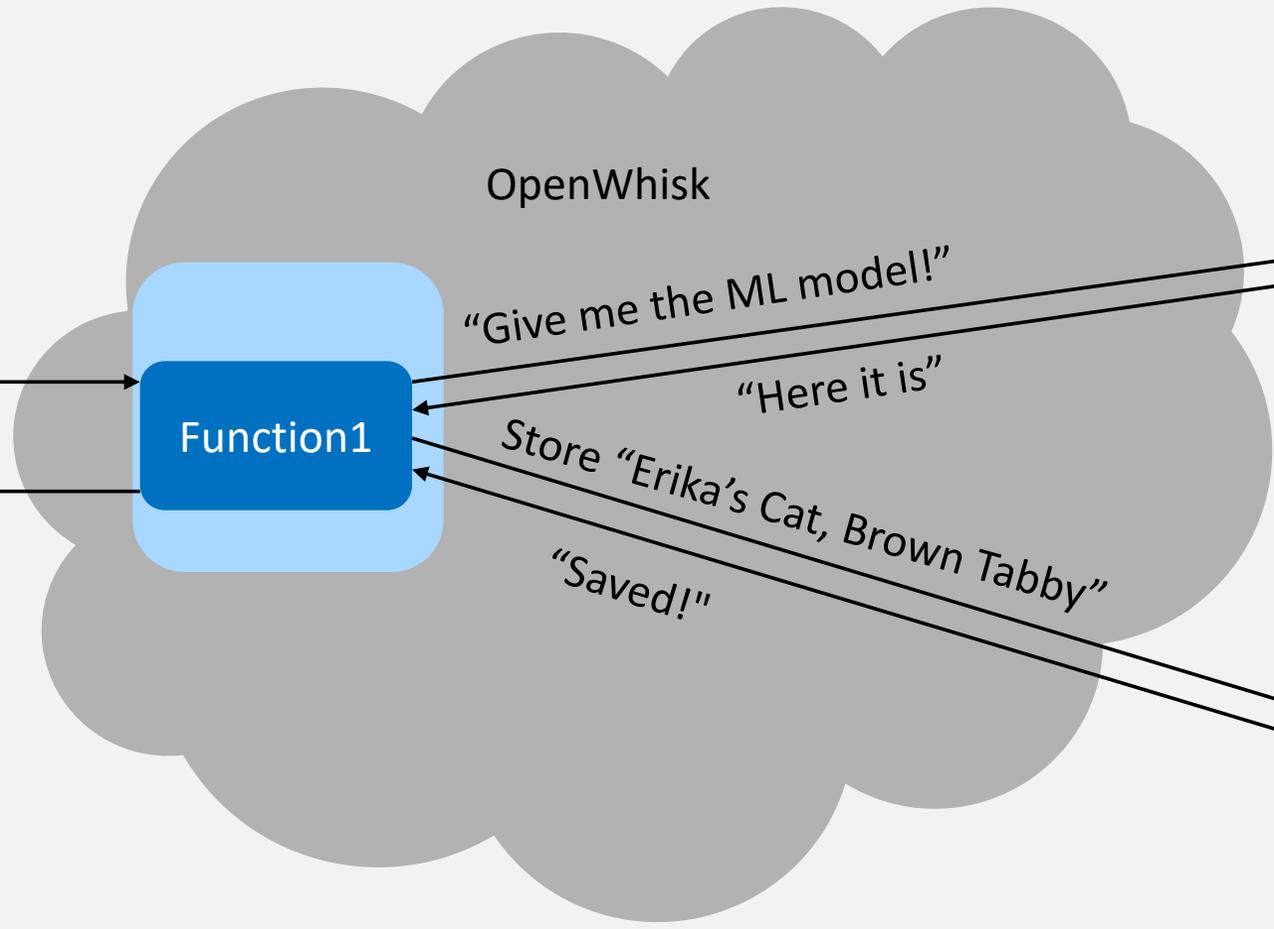
Function Trigger



Function1



Result: "Success!"

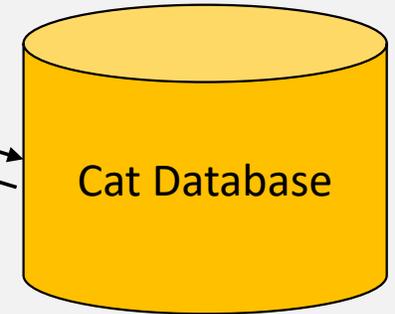
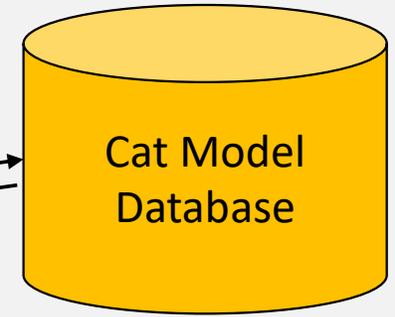


"Give me the ML model!"

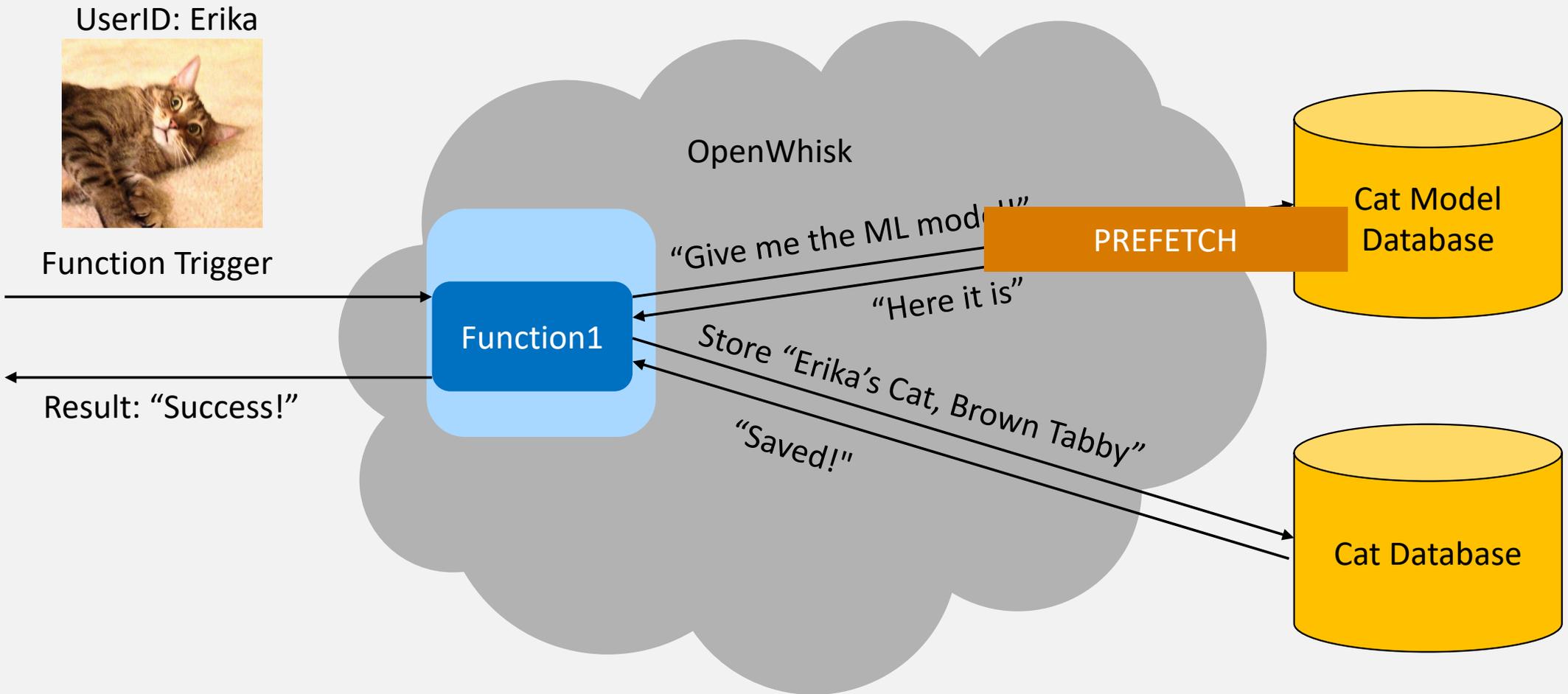
"Here it is"

Store "Erika's Cat, Brown Tabby"

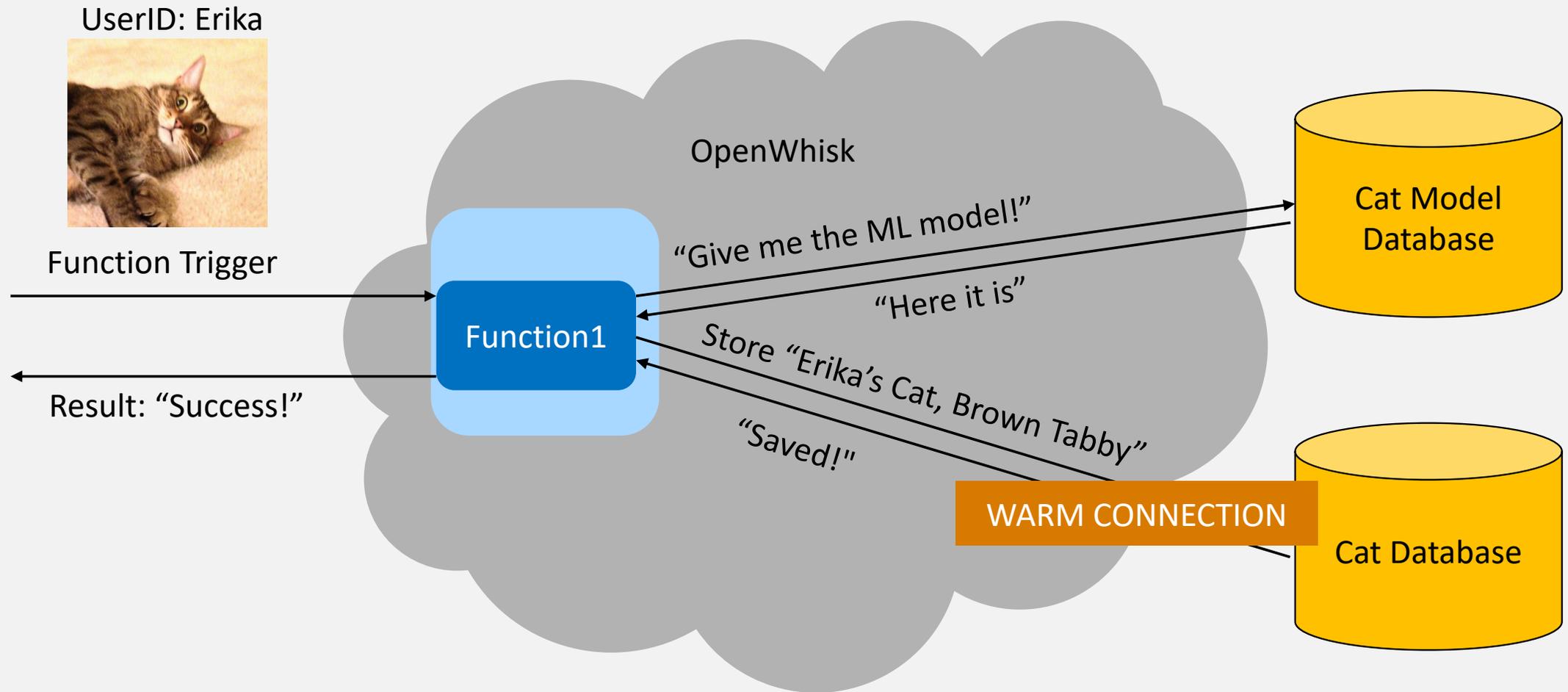
"Saved!"



What Can *Freshen* Do?



What Can *Freshen* Do?



What Can *Freshen* Do?

UserID: Erika



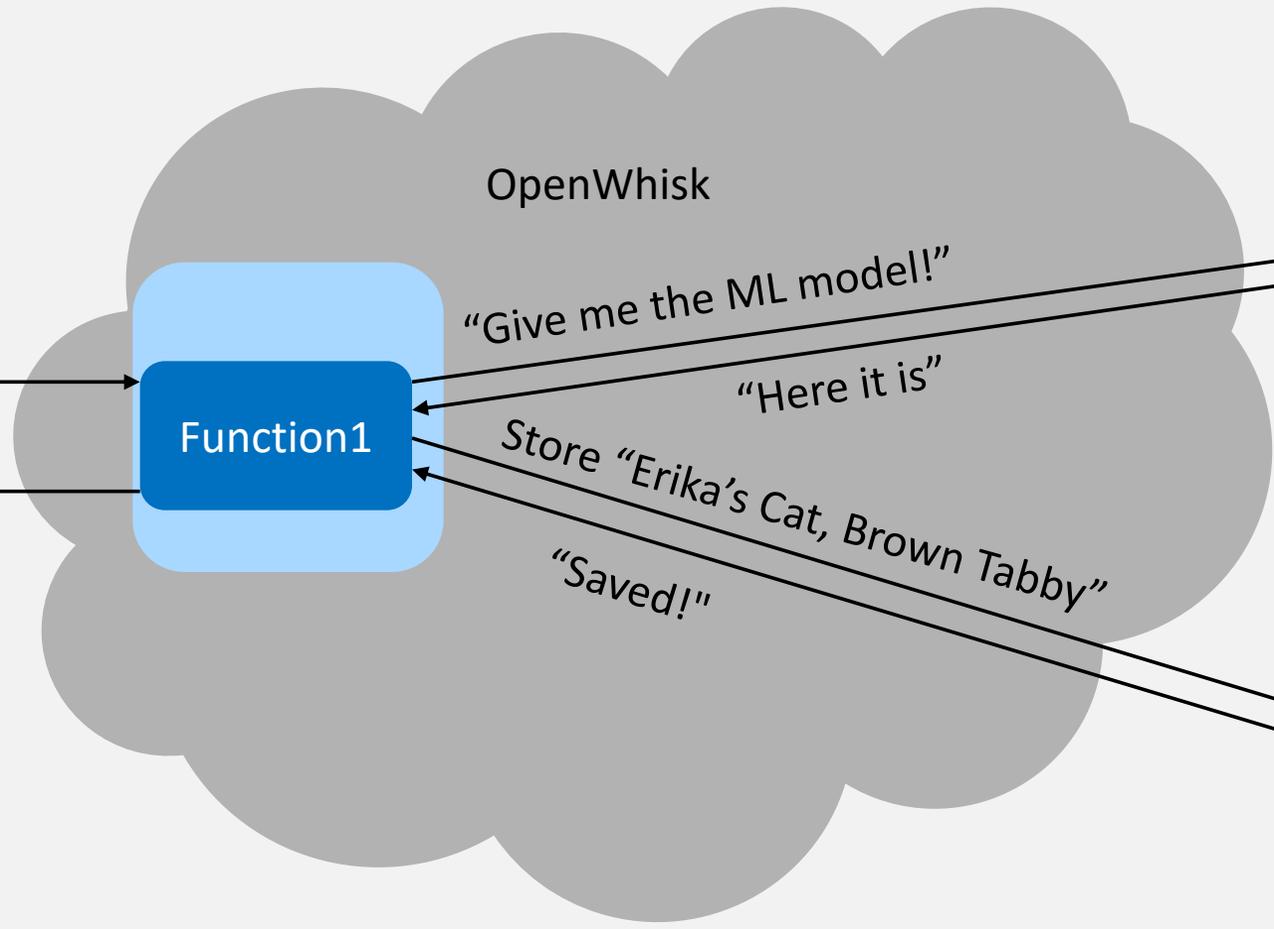
Function Trigger



Function1



Result: "Success!"



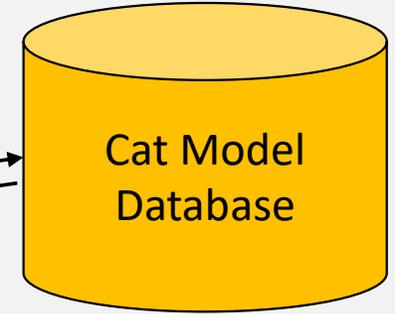
OpenWhisk

"Give me the ML model!"

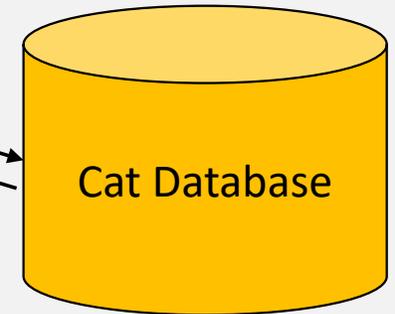
"Here it is"

Store "Erika's Cat, Brown Tabby"

"Saved!"



Cat Model Database



Cat Database

What Can *Freshen* Do?

MORE?

Background

Freshen Design

Evaluation

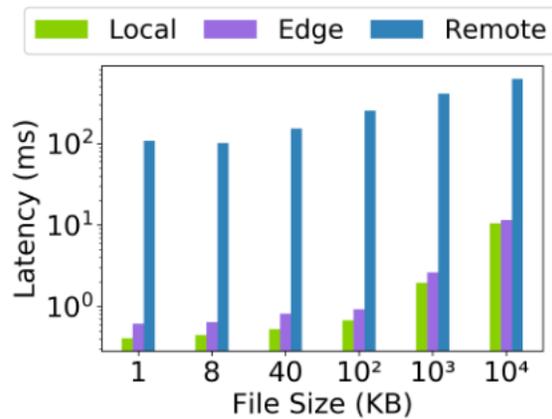
Discussion

Questions

Outline

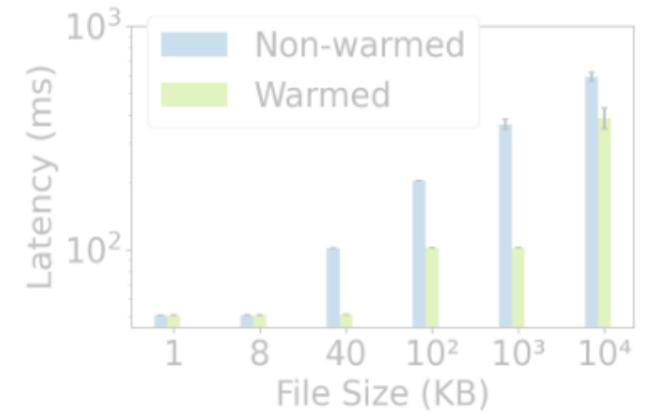
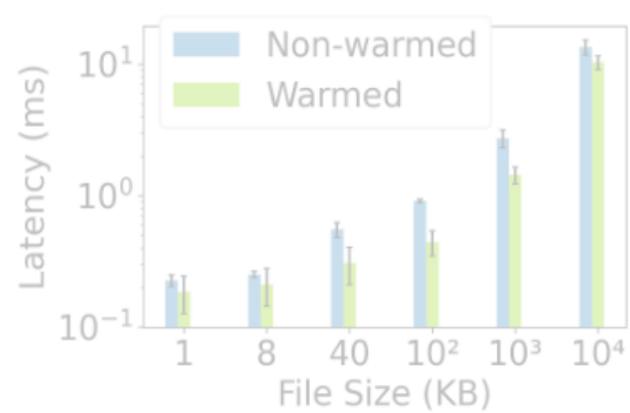
Freshen Motivation

PREFETCH



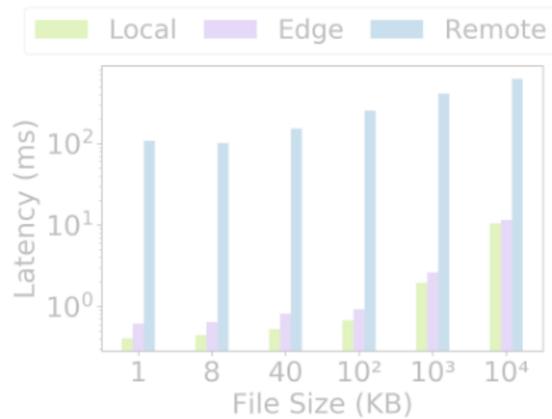
Reduces latency to access data

WARM CONNECTION

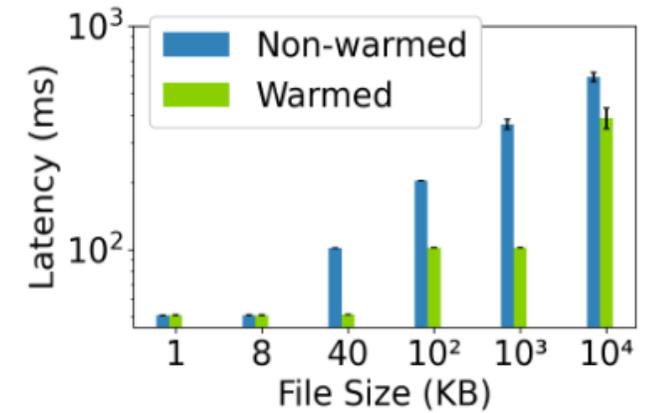
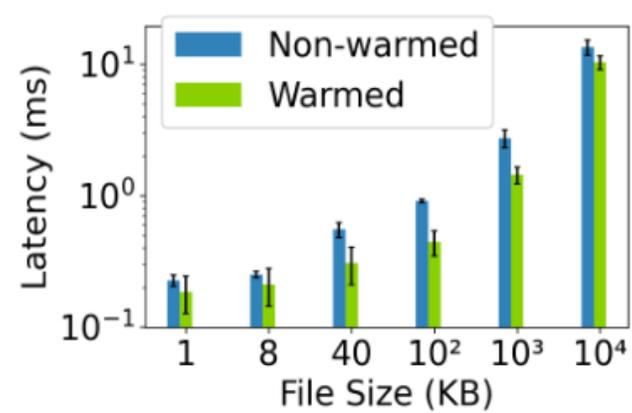


Freshen Motivation

PREFETCH



WARM CONNECTION



TCP connections warmed send traffic more efficiently

Background

Freshen Design

Evaluation

Discussion

Questions

Outline

Discussion

Connection state
manipulation

- How to access?
- Beyond TCP

Function
prediction

Who is
responsible for
freshen?

Other *freshen*
actions

Discussion

Connection state
manipulation

Function
prediction

Who is
responsible for
freshen?

Other *freshen*
actions

Discussion

Connection state
manipulation

Function
prediction

Who is
responsible for
freshen?

- Developer
- Libraries
- Inference

Other *freshen*
actions

Discussion

Connection state
manipulation

Function
prediction

Who is
responsible for
freshen?

Other *freshen*
actions

- Memory allocation?
- Caches?
- Things we have not yet thought of?

Background

Freshen Design

Evaluation

Discussion

Questions

- We propose a new serverless runtime primitive, *freshen*, as a mechanism to enable proactive serverless function resource management.



Erika Hunhoff
erika.hunhoff@colorado.edu