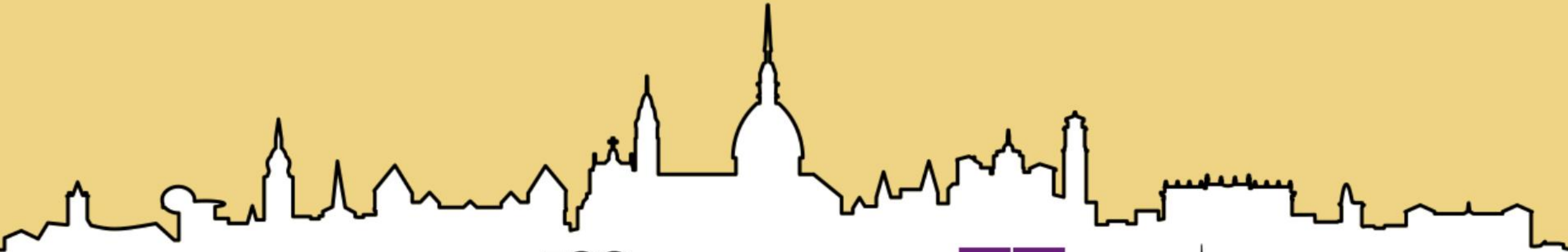
AZURE DAY

Dalle Azure Functions ai Microservizi in Container

Andrea Carratta / Andrè Santacroce @Cloud Ninja



Microsoft msc TECHNOLOGY NET CODE Azure Meetup TORINO TORINO.NET

AZURE DAY

Platinum Sponsor

Microsoft

msc TECHNOLOGY

Technical Sponsor

NET CODE

Azure Meetup TORINO

TORINO .NET

stickermule

Packt›

JET BRAINS

Chi Siamo...

CloudNinja

Andrè Dominic Santacroce
Senior Software Architect

Andrea Carratta
Senior Backend & Cloud Developer

Agenda

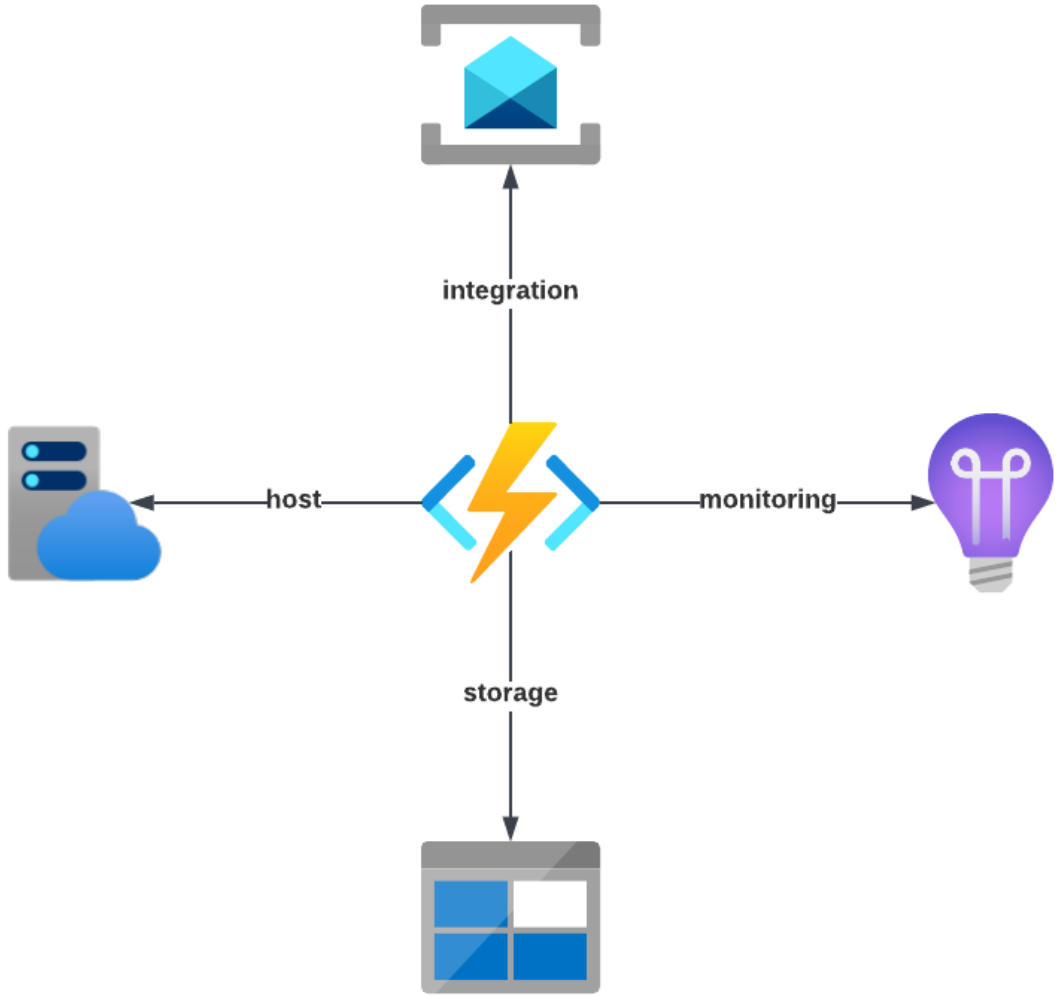
- Serverless models comparison
- Architecture and Infrastructure
- Code examples for common integrations (REST, Cron, Pub-Sub)
- Demo
- Conclusions / Q&A

Azure Functions

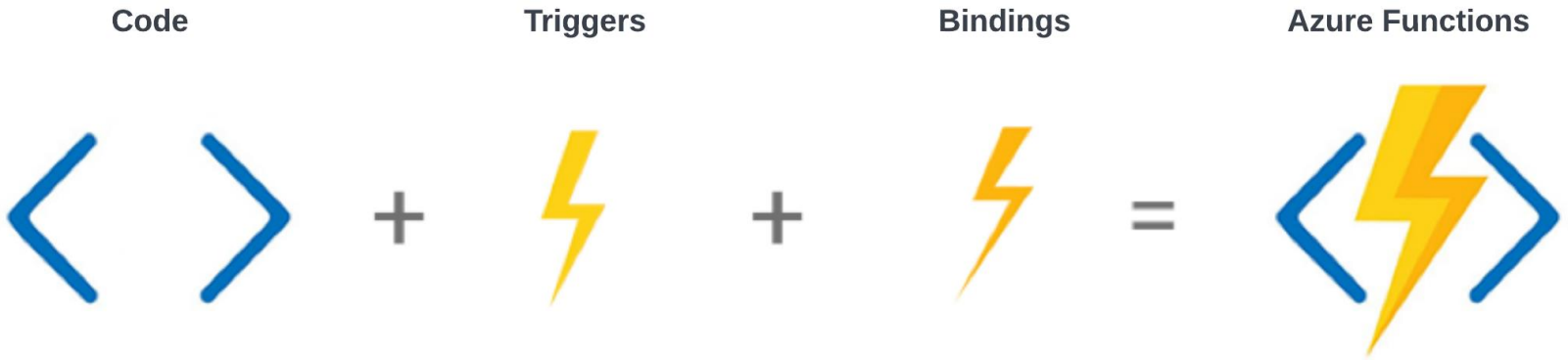
Architettura Azure Functions

Components:

- *Service Bus* to integrate with other services
- *App Service Plan* (Dynamic for Consumption based)
- *Azure Storage* to maintain state
- *App Insights* to support monitoring



Azure Functions Framework



Azure Functions Trigger

HTTP Trigger:

- API exposure
- Synchronous requests
- Easy integration
- Cold start impact

```
ace CloudNinjaFunctionApp.Functions

blic class HTTPTriggerFunction : BaseFunction

  public HTTPTriggerFunction(ILogger<HTTPTriggerFunction> logger) : base(
  {
  }

  [Function(FunctionNames.HTTP_TRIGGER_FUNCTION)]
  public IActionResult Run(
      [HttpTrigger(AuthorizationLevel.Function, "get")] HttpRequest req)
  {
      _logger.LogInformation("C# HTTP trigger function processed a reques
      return new OkObjectResult("Welcome to Azure Functions!");
  }
```

Azure Functions Trigger

CRON Trigger:

- Scheduled executions
- Repetitive tasks
- No input needed
- Workflow automation

```
amespace CloudNinjaFunctionApp

  public class CronTriggerFunction : BaseFunction
  {
      private readonly string[] _topics = null;

      private readonly IServiceBus _serviceBus;

      public CronTriggerFunction(IServiceBus serviceBus, ILoggerFactory loggerFactory, IC

      [Function(FunctionNames.CRON_TRIGGER_FUNCTION)]
      public async Task Run(
          [TimerTrigger("0 */5 * * * *")] TimerInfo myTimer
      )
      {
          _logger.LogInformation($"C# Timer trigger function executed at: {DateTime.Now}"

          var topicsLength = _topics.Length;

          for (int i = 0; i < 200; i++)
          {
              var topicName = _topics[i % topicsLength];
```

Azure Functions Trigger

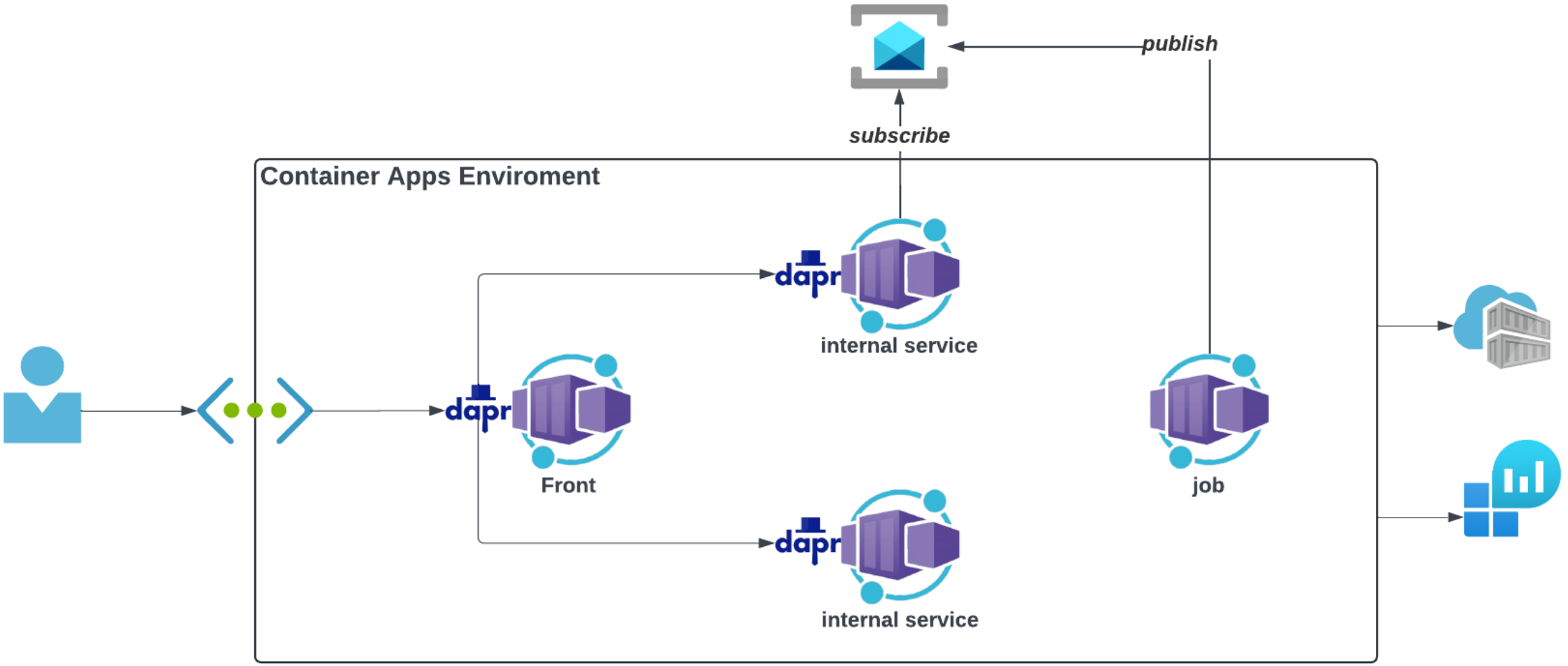
Service Bus Trigger:

- Asynchronous messages
- Event-driven
- High scalability
- Service decoupling

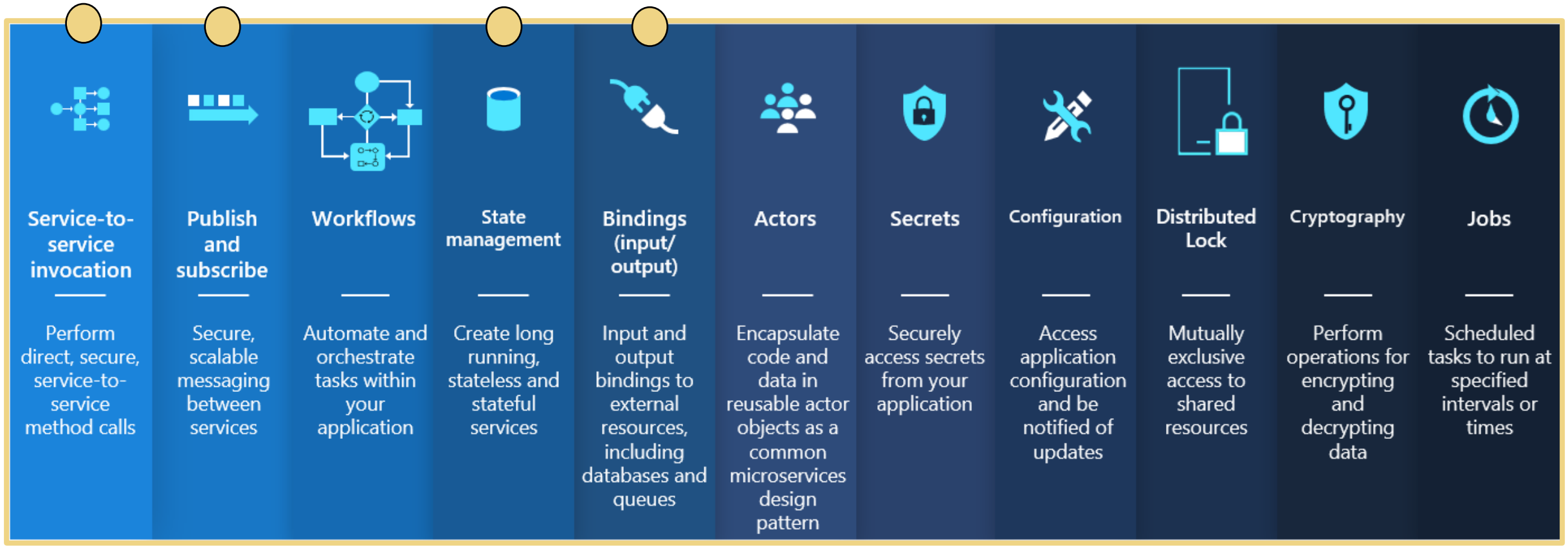
```
public class ServiceBusTriggerFunction
{
    private readonly ILogger<ServiceBusTriggerFunction> _logger;

    public ServiceBusTriggerFunction(ILogger<ServiceBusTriggerFunction> logger)
    {
        _logger = logger;
    }

    [Function(FunctionNames.SERVICE_BUS_TRIGGER_FUNCTION)]
    public async Task Run(
        [ServiceBusTrigger(
                "%ServiceBusTopicNameFunctionApp%",
                "%ServiceBusSubscriptionNameFunctionApp%",
                Connection = "ServiceBusConnectionString")]ServiceBusReceivedMessage message,
          ServiceBusMessageActions messageActions)
    {
        _logger.LogInformation("Message ID: {id}", message.MessageId);
        _logger.LogInformation("Message Body: {body}", message.Body);
        _logger.LogInformation("Message Content-Type: {contentType}", message.ContentType);

        // Complete the message
        await messageActions.CompleteMessageAsync(message);
    }

}
```

Azure Container Apps

Architettura ACA



DAPR Features



DAPR State Component

Key aspects:

- Component ID
- Component Type
- Metadata

main.bicep

```bicep
// Dapr state store component
resource daprComponent 'Microsoft.App/managedEnvironments/daprComponents@2022-03-01' = {
  name: 'statestore'
  parent: environment
  properties: {
    componentType: 'state.azure.blobstorage'
    version: 'v1'
    ignoreErrors: false
    initTimeout: '5s'
    metadata: [
      {
        name: 'accountName'
        value: storageAccountName
      }
      {
        name: 'containerName'
        value: blobContainerName
      }
      {
        name: 'azureClientId'
        value: managedIdentity.properties.clientId
      }
    ]
  }
}
```

ACA Hello API

Key aspects:

- DAPR Client
- Store ID
- Read / Write / Delete
- OCC Concurrency

Program.cs

CloudNinja.DaprHello

```
22
23      app.MapGet("/hello-state", async () =>
24      {
25          var client = new DaprClientBuilder().Build();
26
27          // Get state from the state store
28          var result = await client.GetStateAsync<int>("statestore", "myintkey");
29
30          // Save state into the state store
31          await client.SaveStateAsync("statestore", "myintkey", ++result);
32
33          return result;
34      })
35      .WithName("HelloState")
36      .WithOpenApi();
```

DAPR PubSub Component

Key aspects:

- Scalability configuration
- Max message per replica
- Max throughput

infra > services > aca-env > main.bicep > {} daprPubSubComponent

```
109  // Dapr pubsub component for Azure Service Bus
110  resource daprPubSubComponent 'Microsoft.App/managedEnvironments/daprComponents@2022-03-01' = {
111    name: 'pubsub'
112    parent: environment
113    properties: {
114      componentType: 'pubsub.azure.servicebus'
115      version: 'v1'
116      ignoreErrors: false
117      initTimeout: '5s'
118      metadata: [
119        {
120          name: 'connectionString'
121          value: serviceBusAuthRule.listKeys().primaryConnectionString
122        }
123        {
124          name: 'maxActiveMessages'
125          value: '100'
126        }
127        {
128          name: 'maxConcurrentHandlers'
129          value: '20'
130        }
131      ]
132    }
133  }
```

DAPR Publish-Subscribe

Key aspects:

- Pub/Sub Component
- Input Topic Binding

Program.cs

CloudNinja.DaprConsumer

```
15
16      // Dapr subscription in [Topic] routes orders topic to this route
17      app.MapPost("/process", [Topic("pubsub", "test")] async (object message) => {
18
19          Console.WriteLine("Received : " + JsonSerializer.Serialize(message));
20
21          var envReplica = Environment.GetEnvironmentVariable("CONTAINER_APP_REPLICA_NAME");
22          var replica = string.IsNullOrEmpty(envReplica) ? "default-replica" : envReplica;
23          await Task.Delay(10000);
24
25          Console.WriteLine($"{replica} consumed topic message.");
26
27          return Results.Ok();
28      });
29
```

DAPR Service-to-Service

Key aspects:

- HttpClient by appId
- Standard REST Invocation
- DAPR SideCar

Program.cs

CloudNinja.DaprHello

```
37
38  app.MapGet("/invoke-dapr", async (int count) =>
39  {
40      var client = DaprClient.CreateInvokeHttpClient(appId: "daprconsumer");
41
42      Stopwatch watch = new Stopwatch();
43      watch.Start();
44      for (int i = 0; i < count; i++)
45      {
46          // Invoking a service
47          var response = await client.PostAsJsonAsync("/process-dapr", i);
48
49          Console.WriteLine("processed: " + i);
50      }
51      watch.Stop();
52
53      Console.WriteLine($"Time elapsed: {watch.ElapsedMilliseconds} ms");
54
55      return watch.ElapsedMilliseconds;
56  })
57  .WithName("InvokeDapr")
58  .WithOpenApi();
```

Container App Definition

Environment / Ingress / DAPR

infra > modules > aca > app.bicep > { } coreApi

```
42  resource coreApi 'Microsoft.App/containerApps@2022-03-01' = {
52    properties: {
53      managedEnvironmentId: acaEnvironment.id
54      configuration: {
55        activeRevisionsMode: 'multiple'
56        registries: [
57          {
58            server: acrFullName
59            identity: managedIdentity.id
60          }
61        ]
62        ingress: {
63          external: ingressEnabled
64          targetPort: 8080
65        }
66        dapr: {
67          enabled: true
68          appId: appId
69          appProtocol: 'http'
70          appPort: 8080
71        }
72        secrets: secrets
73      }
```

Container / Resources / Scalability

```
74      template: {
75        revisionSuffix: imageTag
76        containers: [
77          {
78            image: imageName
79            name: appId
80            env: enviromentVariables
81            resources: {
82              cpu: json(cpu)
83              memory: '${memory}Gi'
84            }
85          }
86        ]
87        scale: scale
88      }
89    }
90  }
```

KEDA Scaler

Key aspects:

- Zero Instance Scale
- Rule Triggers
- Observed event stream

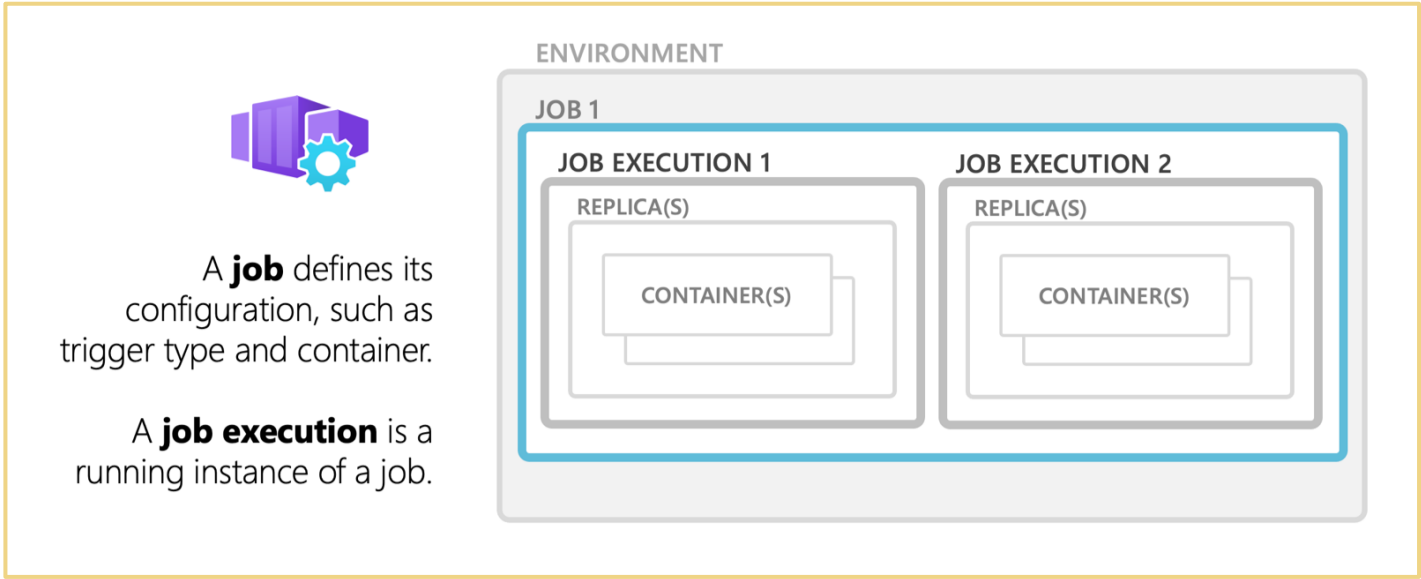
infra > services > dapr-consumer > main.bicep > {} acaApp

```
59   module acaApp '../../modules/aca/app.bicep' = {
101      scale: {
102        minReplicas: minReplica
103        maxReplicas: maxReplica
104        rules: [
105          {
106            name: 'topic-based-scaling'
107            custom: {
108              type: 'azure-servicebus'
109              identity: 'user-assigned'
110              metadata: {
111                topicName: topicName
112                subscriptionName: 'daprconsumer'
113                messageCount: scalerMessageCount
114              }
115              auth: [
116                {
117                  secretRef: serviceBusAuthorizationSecret
118                  triggerParameter: 'connection'
119                }
120              ]
121            }
122          }
123        ]
124      }
125    }
126  }
```

ACA Jobs

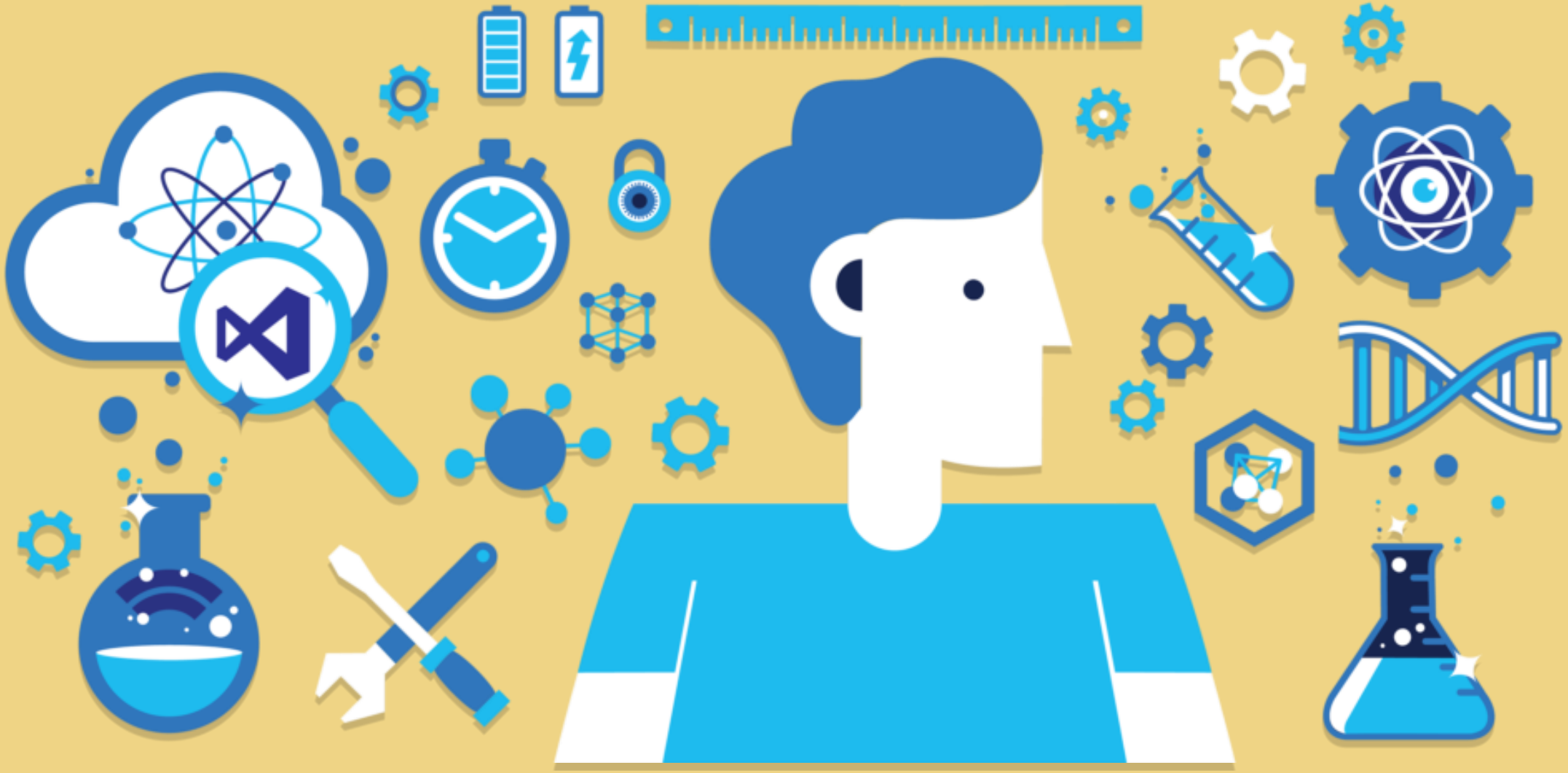
Key aspects:

- Runs within the ACA Environment
- Unit of work defined as the command to run on the Published Container
- Single or parallel excecution by using Replicas
- Different Types of triggers: Manual, Schedule, Event



AZURE DAY

Demo



Conclusions and Q&A

	Azure Functions	Azure Container Apps
Excecution Model	Function-as-a-service	Container-as-a-service
Programming Model	Azure Functions Framework. Fast learning curve, coupled framework evolution/version.	Full run time control, custom frameworks and libraries may be included for what can be Containerized. Built-in DAPR support.
Scalability	Out-of-the-box Event based scaler, no specific configuration required	Advanced scalability control through KEDA
Cold Start	On Consumption plan strong impact. Still smaller than ACA. Using Dedicated/Flex plans we gain control.	With scale to zero very slow startup. With single inactive we obtain a good balance.
Best Use Case	Short-lived event driven tasks. Shines on simple integrations.	Microservices architecture. Suitable to sustain an overall platform.

Contatti

CloudNinja

https://cloudninja.it

Andrè Dominic Santacroce
Senior Software Architect
card.cloudninja.it/andre-santacroce

Andrea Carratta
Senior Backend & Cloud Developer
card.cloudninja.it/andrea-carratta

Thank You!

Vote my session

AZURE DAY

Platinum Sponsor

Microsoft

msc TECHNOLOGY

Technical Sponsor

NET CODE

Azure Meetup TORINO

TORINO .NET

stickermule

Packt›

JET BRAINS