Uma Introdução ao Actor Model com Microsoft Orleans

THE DEVELOPER'S CONFERENCE

2019 – Trilha .NET

Agenda

1. Introdução
2. Microsoft Orleans e o Actor Model
3. Arquitetura stateful
4. Código e Demo

Quem sou eu?

Fabio Gouw

Arquiteto de Soluções @ Itaú

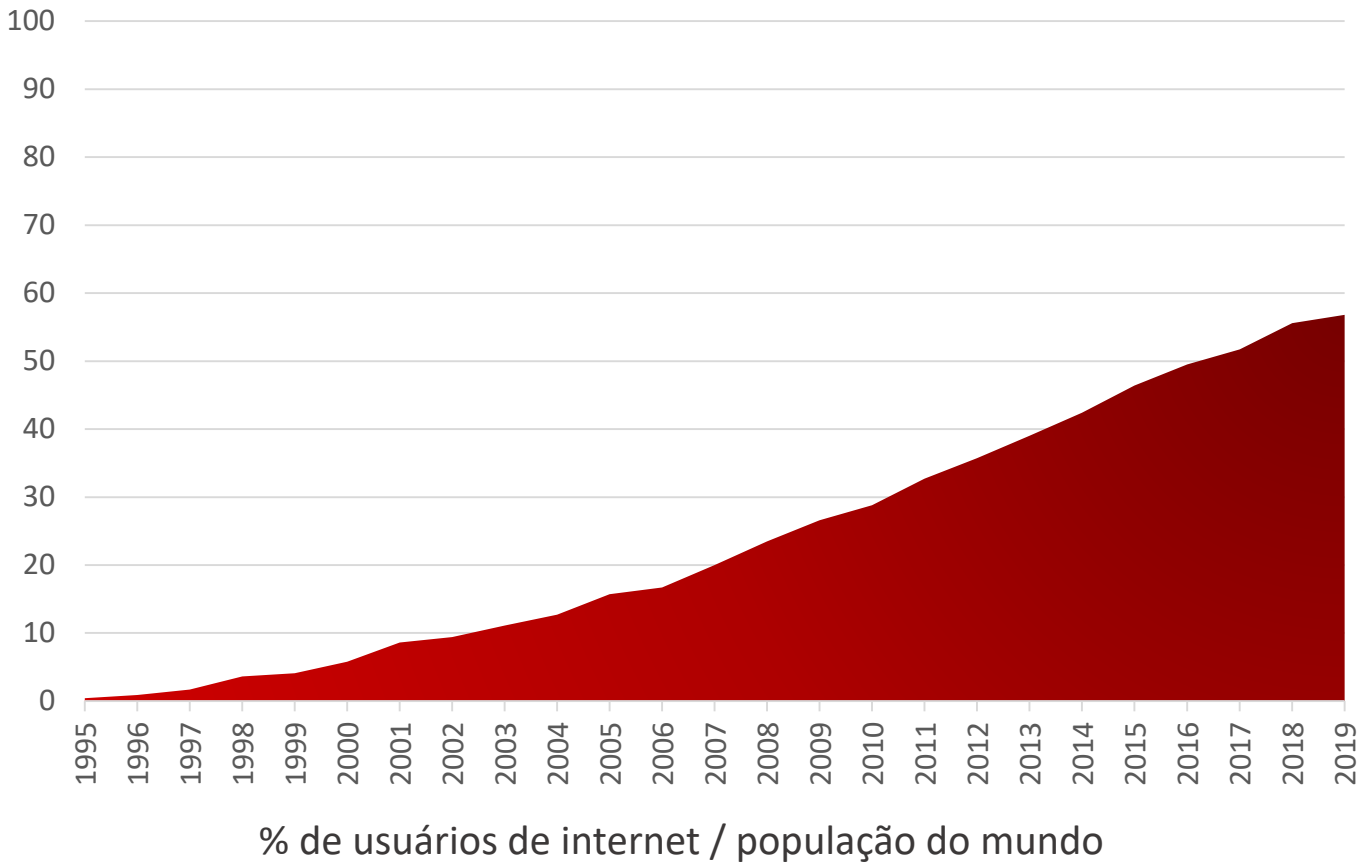
fabiogouw

@fabiogouw

http://www.fabiogouw.com

Introdução

Introdução | Um pouco de estatística...



% de usuários de internet / população do mundo

Em um minuto...

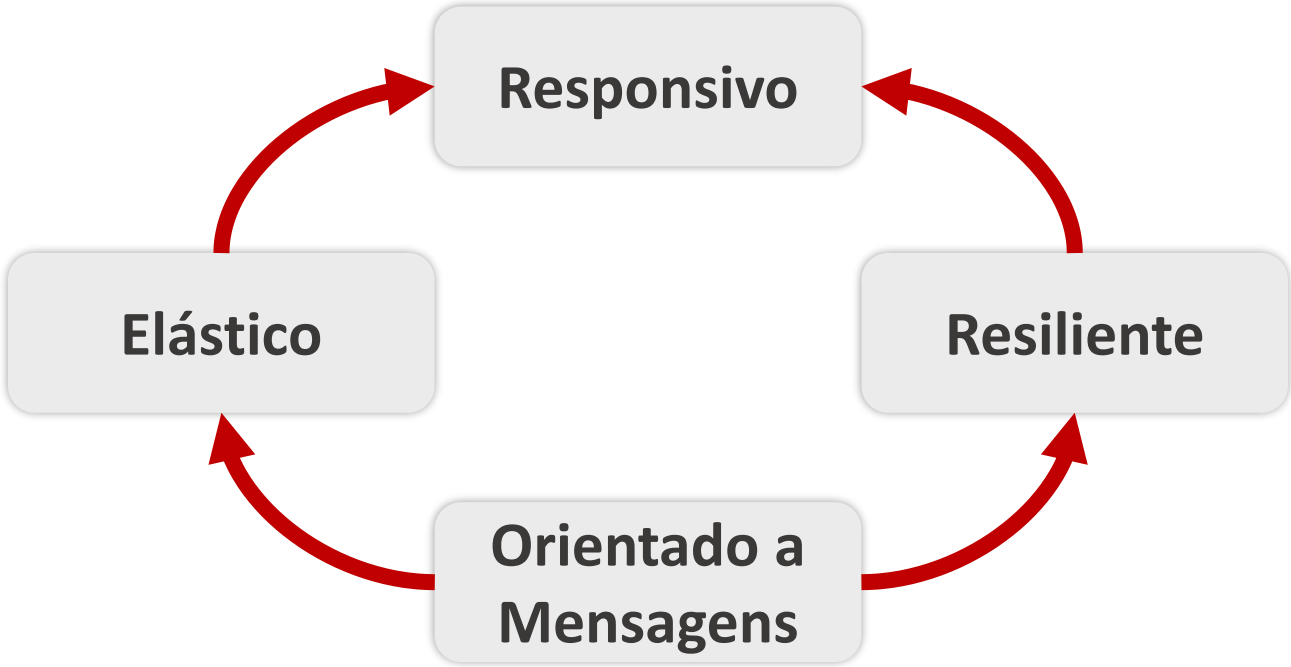
1 milhão
de logins

29 milhões
de mensagens

3,8 milhões
de buscas

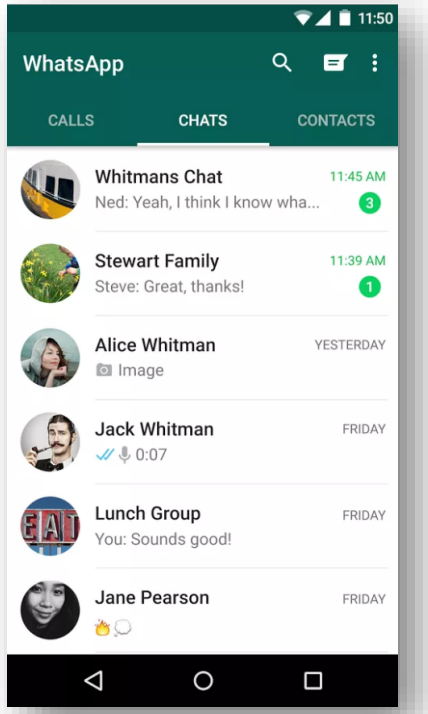
188 milhões
de e-mails

Introdução | Sistemas Reativos



https://www.reactivemanifesto.org/

Introdução | Exemplos de Aplicações



akka

ERLANG

Orleans

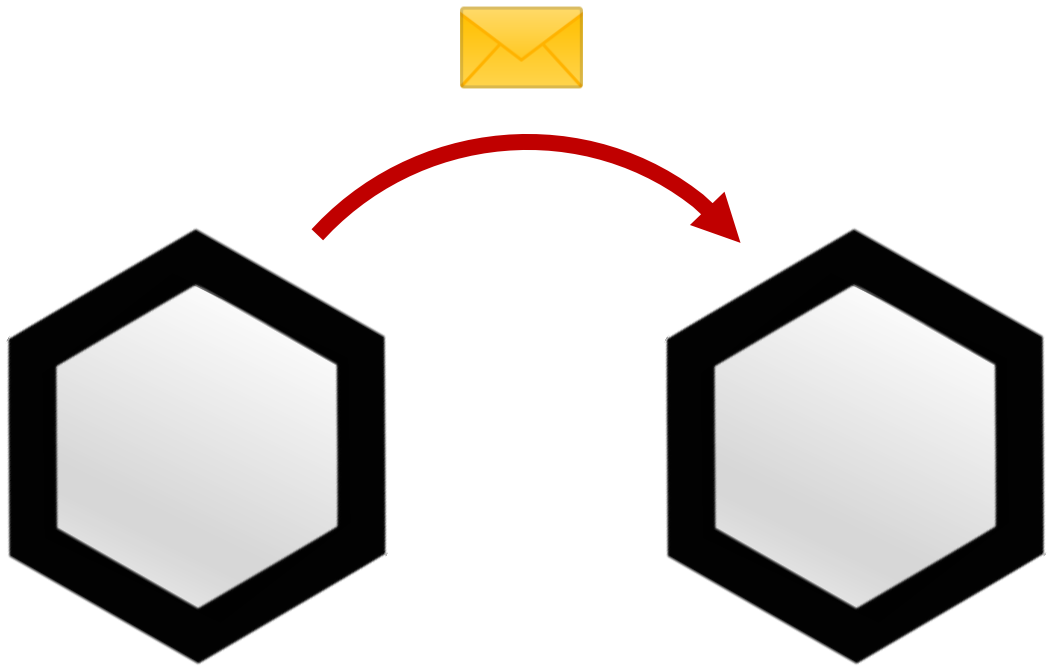
Microsoft Orleans

Orleans | Framework e história

Framework para criação de **aplicações distribuídas** baseado no **Actor Model**, voltado para cloud, em cenários **stateful** e near **real-time**.

Actor Model | Conceitos

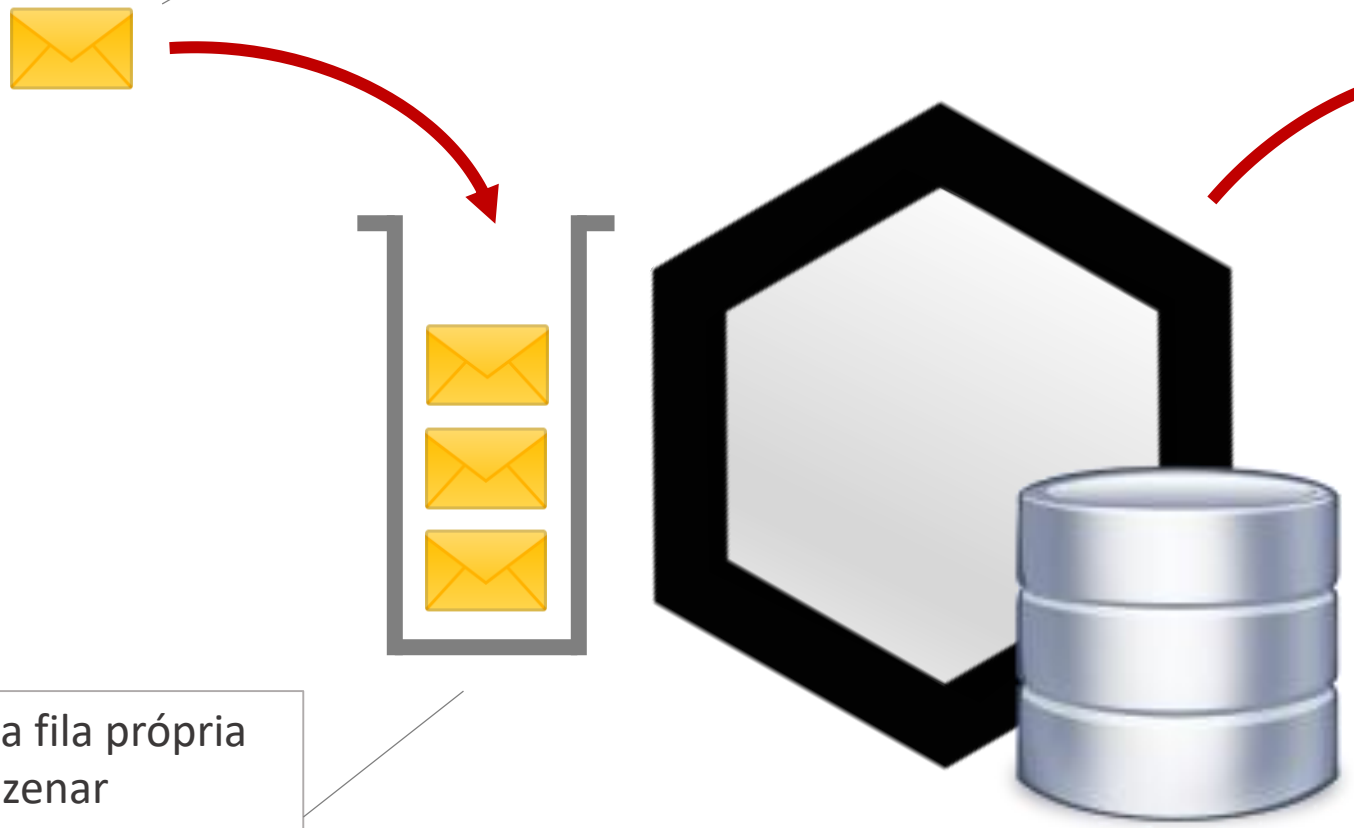
Modelo conceitual de **processamento concorrente** que trata atores como sua primitiva de computação.



Atores...

- representam conceitos da vida real
- possuem identidade única
- se comunicam através de mensagens

Actor Model | Conceitos



Recebe as mensagens de forma assíncrona

Envia mensagens para outros atores

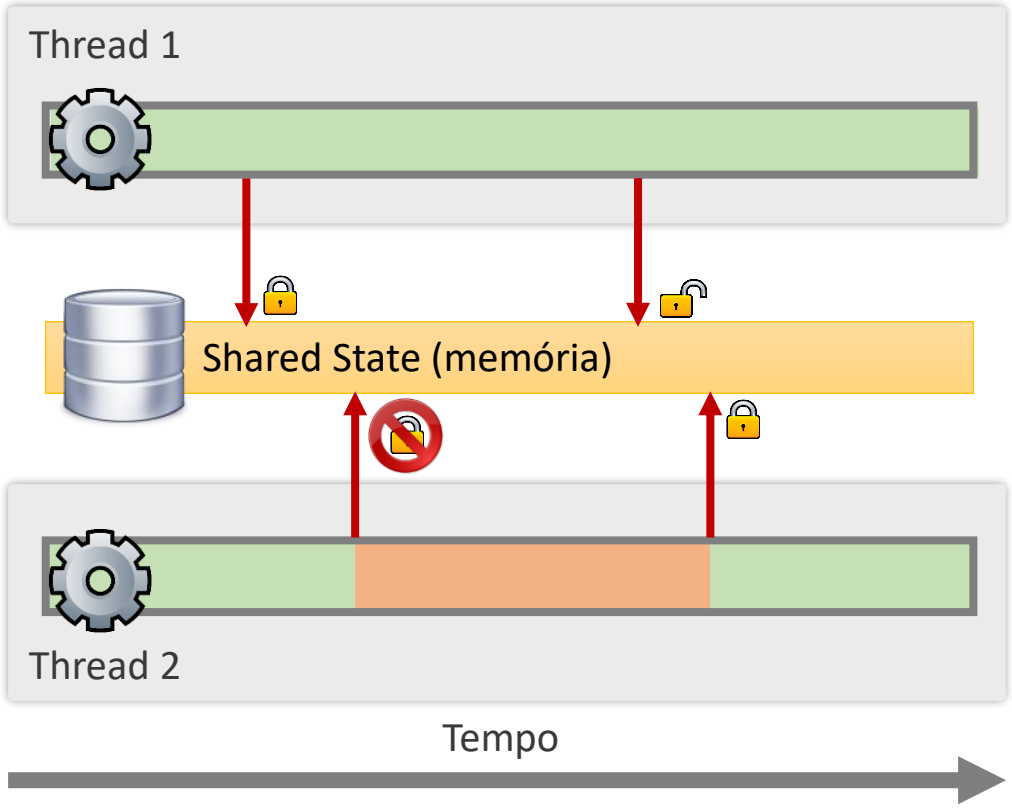
Processamento é feito de forma single-threaded, mensagem a mensagem

Possui uma fila própria para armazenar mensagens a processar (mailbox)

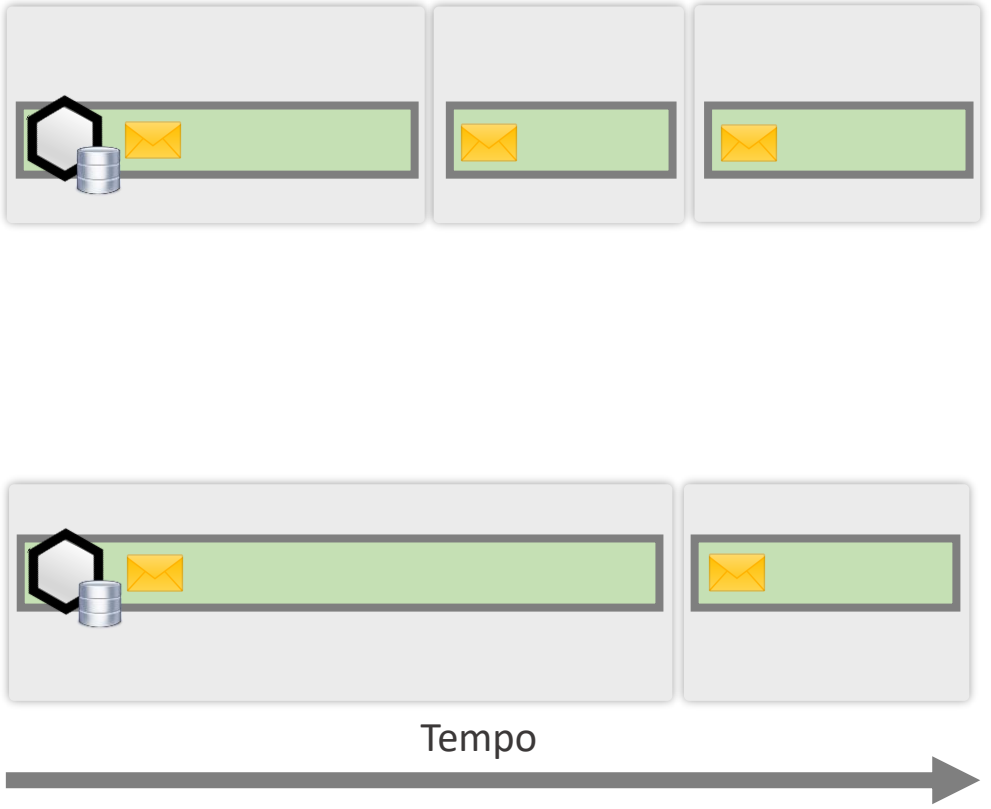
Cada ator possui seu próprio estado interno

Actor Model | Controle de Concorrência

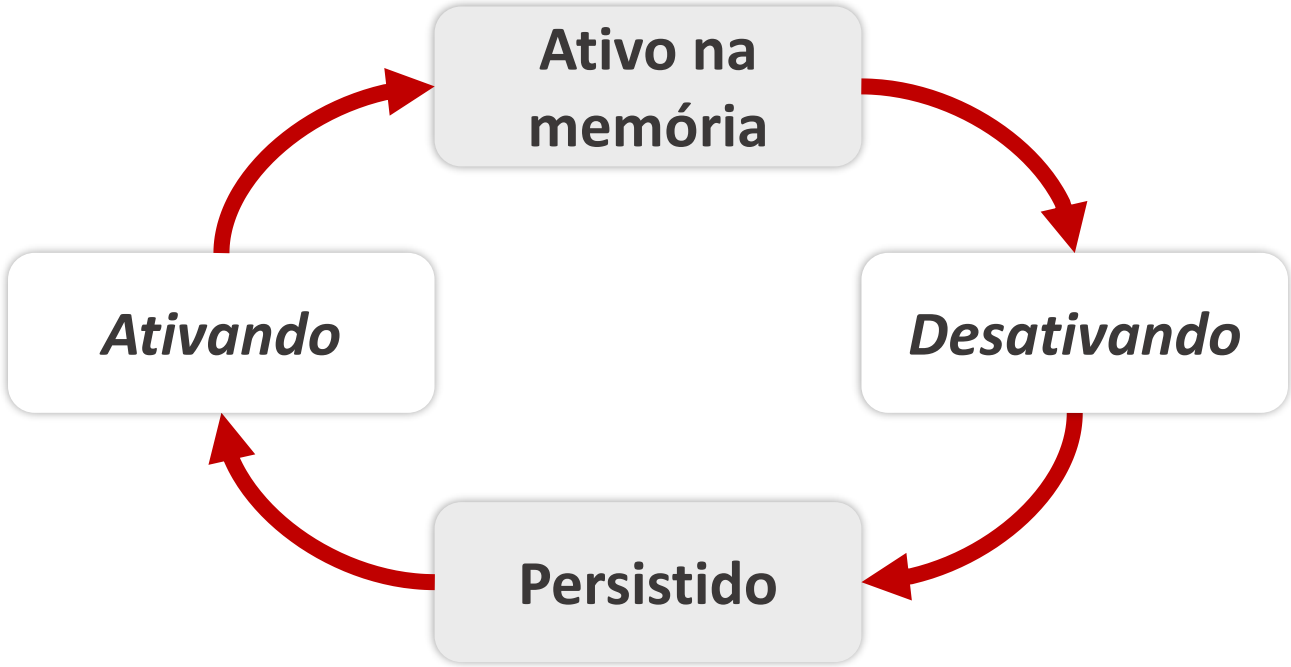
com uso de locks



orientado a mensagens



Orleans | Ativação Transparente



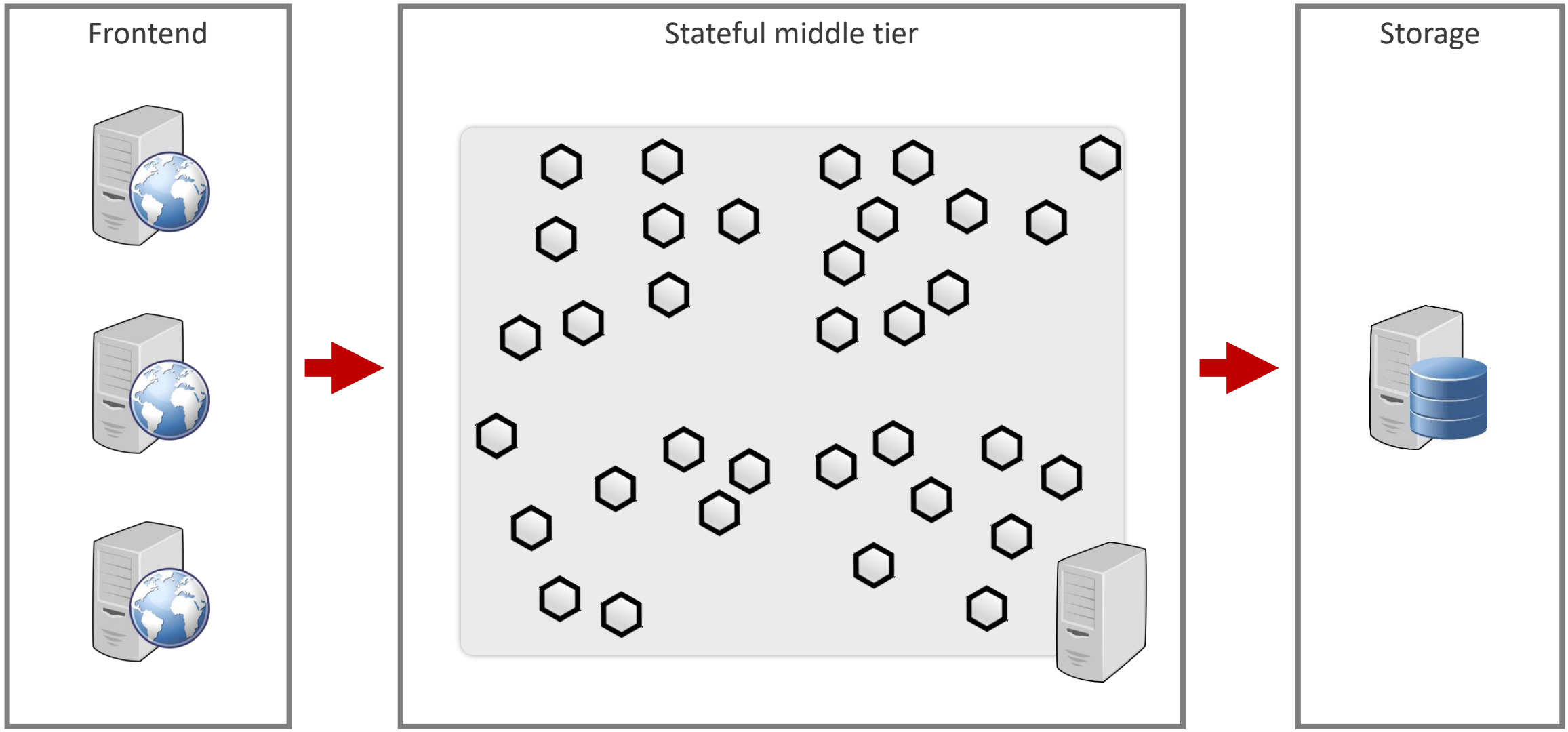
Orleans

Virtual Actor → Grains

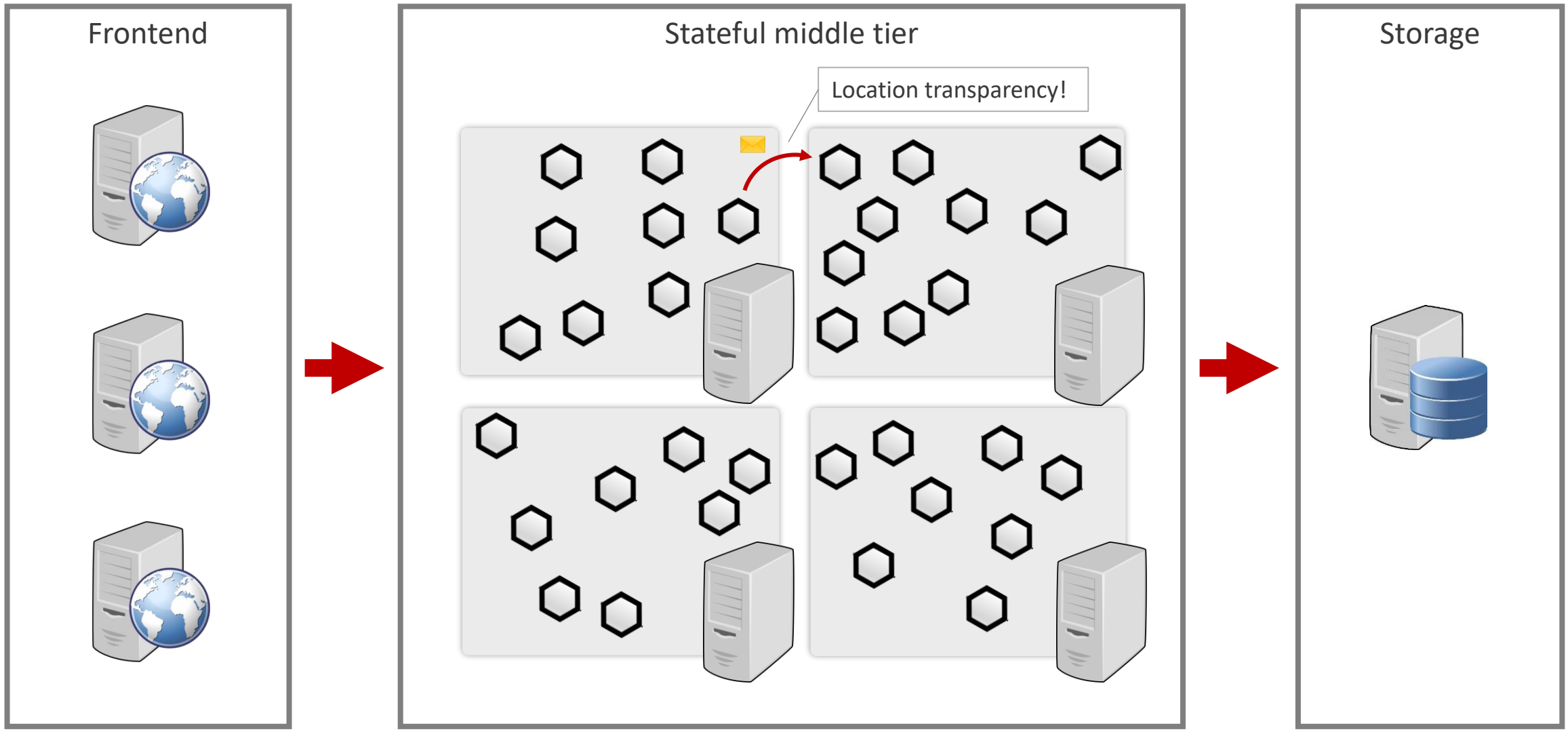
Host de Grains → Silos

Arquitetura com o Microsoft Orleans

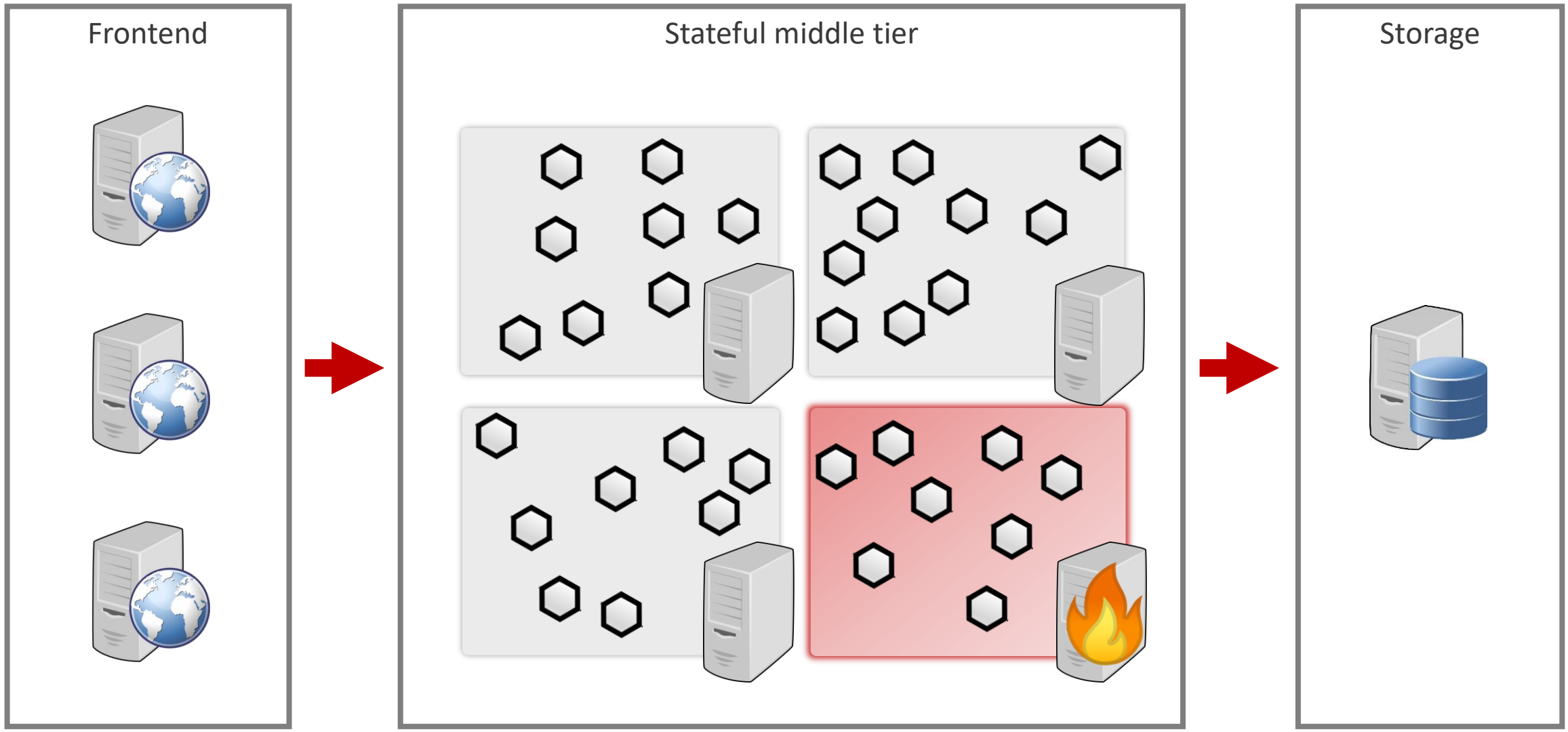
Orleans | Arquitetura



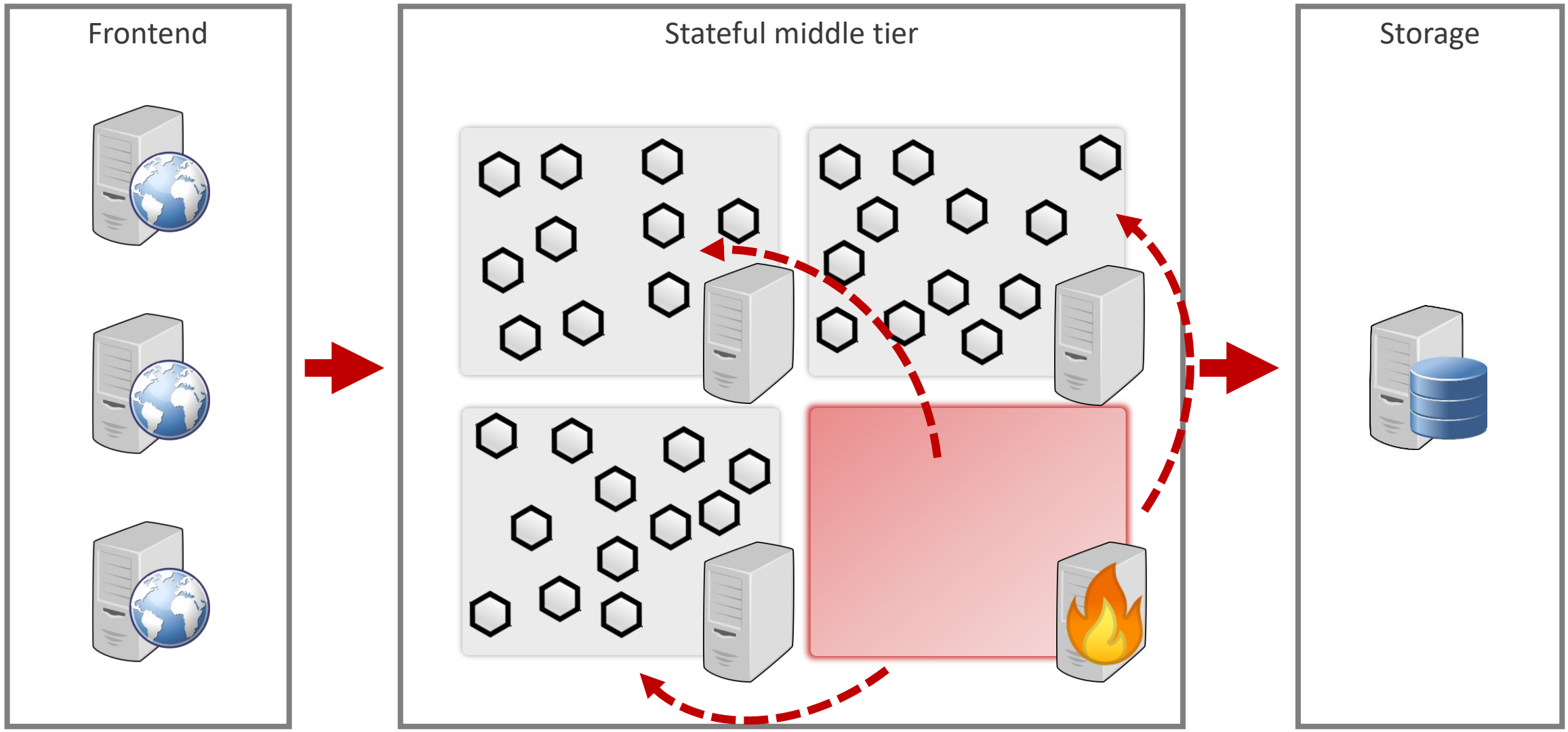
Orleans | Arquitetura



Orleans | Arquitetura

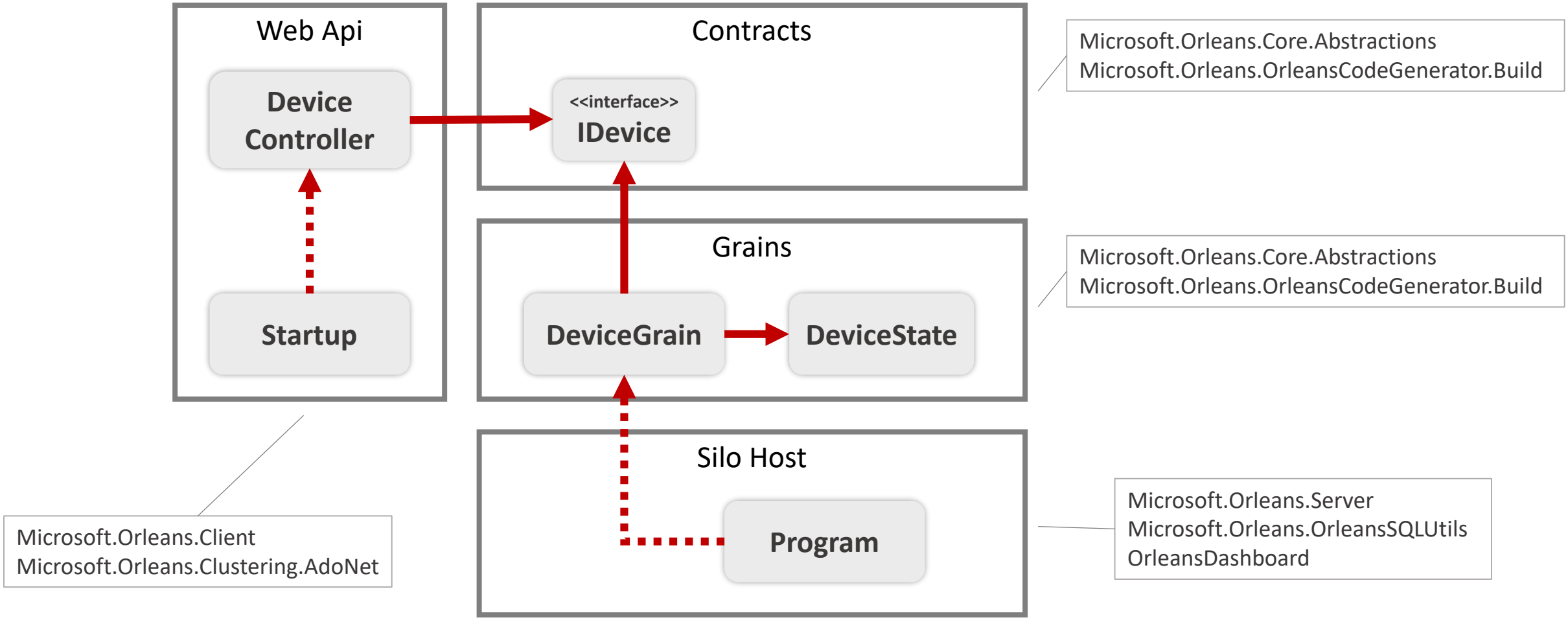


Orleans | Arquitetura

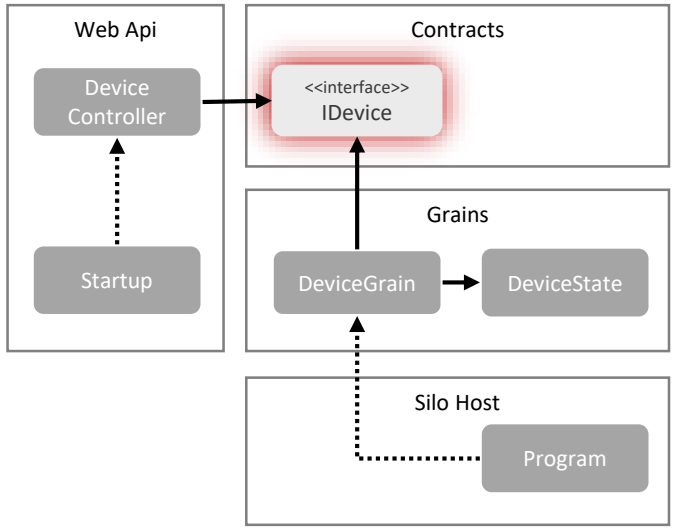


Exemplo de Código

Exemplo | Componentes e Classes

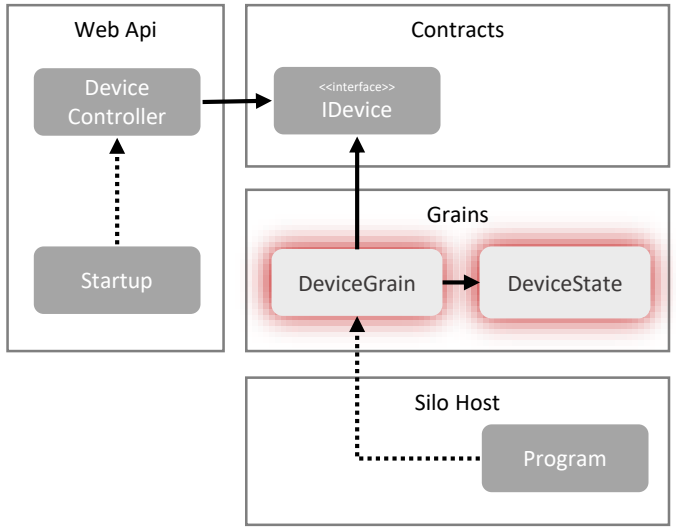


Exemplo | Interface – contrato único




```
public interface IDevice : IGrainWithIntegerKey
{
    Task SetTemperature(double temperature);
    Task<double> GetTemperature();
}
```

Exemplo | Implementação do ator (grain)

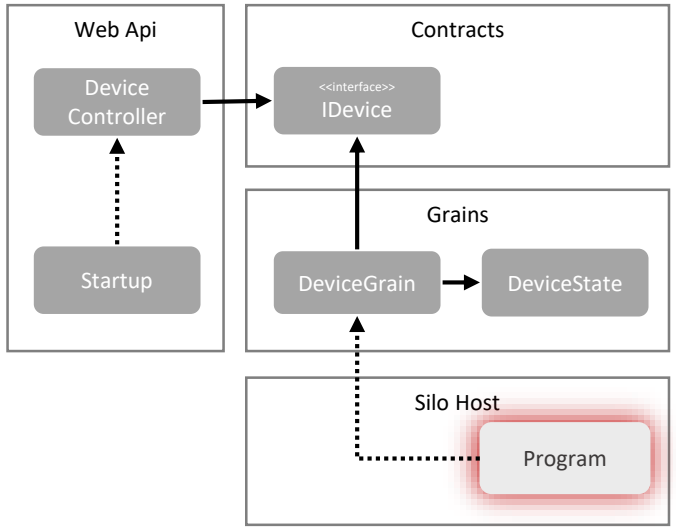



```
[StorageProvider(ProviderName="Devices")]
public class DeviceGrain : Grain<DeviceState>, IDevice
{
    public Task<double> GetTemperature()
    {
        return Task.FromResult(State.LastTemperature);
    }

    public async Task SetTemperature(double temperature)
    {
        State.LastTemperature = temperature;
        await WriteStateAsync();
    }
}

public class DeviceState
{
    public double LastTemperature {get;set;}
}
```

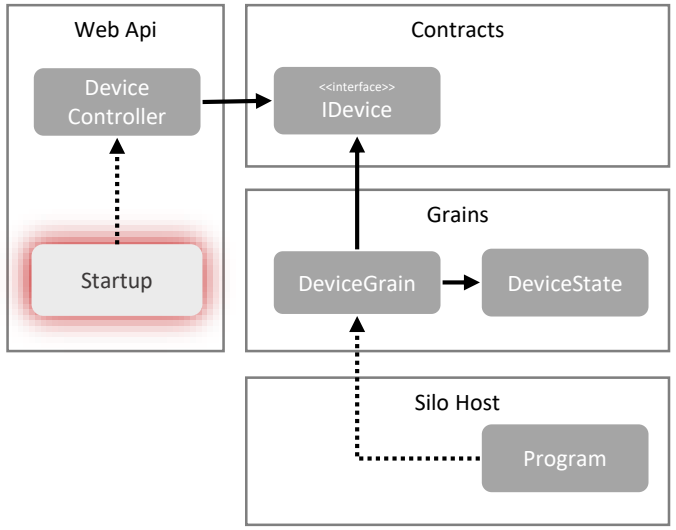
Exemplo | Host dos atores (server)




```
public static int Main(string[] args)
{
    ...
    var builder = new SiloHostBuilder()
    .Configure<ClusterOptions>(options => {
        options.ClusterId = "dev";
        options.ServiceId = "OrleansDemo";
    })
    .UseAdoNetClustering(options => {
        options.Invariant = "System.Data.SqlClient";
        options.ConnectionString = ...
    })
    .AddAdoNetGrainStorage("Devices", options=> {
        options.Invariant = "System.Data.SqlClient";
        options.ConnectionString = ...
    })
    .ConfigureEndpoints(siloPort: 11111, gatewayPort: 30000)
    .UseDashboard(_ => {  })
    .ConfigureApplicationParts(parts => parts.AddFromApplicationBaseDirectory())
    .ConfigureLogging(logging => logging.AddConsole());

    var host = builder.Build();
    await host.StartAsync();
}
```

Exemplo | Configuração do client

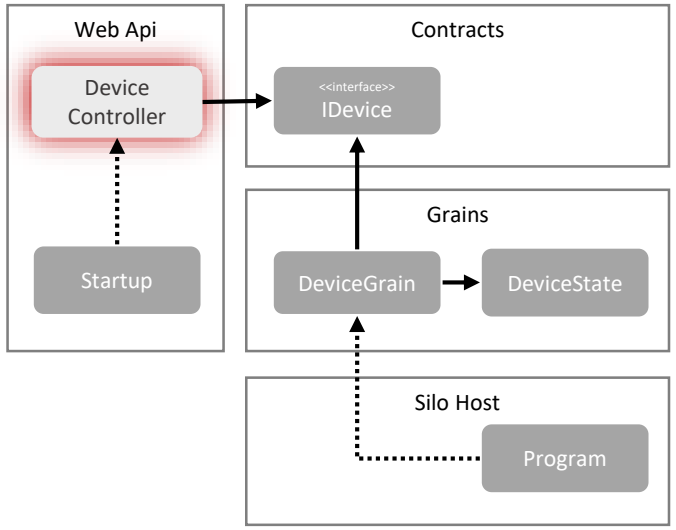



```
// Startup.cs
IClusterClient client = new ClientBuilder()
    .Configure<ClusterOptions>(options =>
    {
        options.ClusterId = "dev";
        options.ServiceId = "OrleansDemo";
    })
    .UseAdoNetClustering(options =>
    {
        options.Invariant = "System.Data.SqlClient";
        options.ConnectionString = ...
    })
    .ConfigureLogging(logging => logging.AddConsole())
    .Build();
await client.Connect();

...

services.AddSingleton(client);
```

Exemplo | Web API de consumo




```
[Route("api/[controller]")]
[ApiController]
public class DeviceController : ControllerBase {
    private readonly IClusterClient _client;

    public DeviceController(IClusterClient client) {
        _client = client;
    }

    // GET api/device/5
    [HttpGet("{deviceId}")]
    public async Task<double> Get(int deviceId) {
        var device = _client.GetGrain<IDevice>(deviceId);
        return await device.GetTemperature();
    }

    // PUT api/device/5
    [HttpPut("{deviceId}")]
    public async Task Put(int deviceId, [FromForm] string value) {
        var device = _client.GetGrain<IDevice>(deviceId);
        await device.SetTemperature(double.Parse(value));
    }
}
```

~~Live~~ Demo

https://youtu.be/Q-JkcgoqjBo

Conclusão | Para saber mais

- Exemplo https://github.com/fabiogouw/OrleansDemo
- Documentação do Microsoft Orleans https://dotnet.github.io/orleans/
- Vídeo *Building the Halo 4 Services with Orleans* - Caitie McCaffrey https://www.infoq.com/presentations/halo-4-orleans/
- Livro *Reactive Messaging Patterns with the Actor Model: Applications and Integration in Scala and Akka* - Vaughn Vernon
- Post *Design Techniques for Building Stateful, Cloud-Native Applications* – Hugh McKee https://www.lightbend.com/blog/design-techniques-stateful-cloud-native-applications-resilience-recoverability-scalability

Obrigado!

 fabiogouw

 @fabiogouw

 http://www.fabiogouw.com