



Edição 2025

# Aplicações Web reativas com Blazor SSR e SignalR



Ano de 2025

[contato@mvpconf.com.br](mailto:contato@mvpconf.com.br)

# Nossos Patrocinadores:



# Gustavo Bigardi

*CTO & Consultor @ Clarus IT*

Microsoft MVP 6x consecutivas;  
Desenvolvedor de Software a  
mais de 20 anos;  
Trabalho com diversas  
tecnologias, sendo .NET minha  
stack principal;

<https://about.me/gbbigardi>



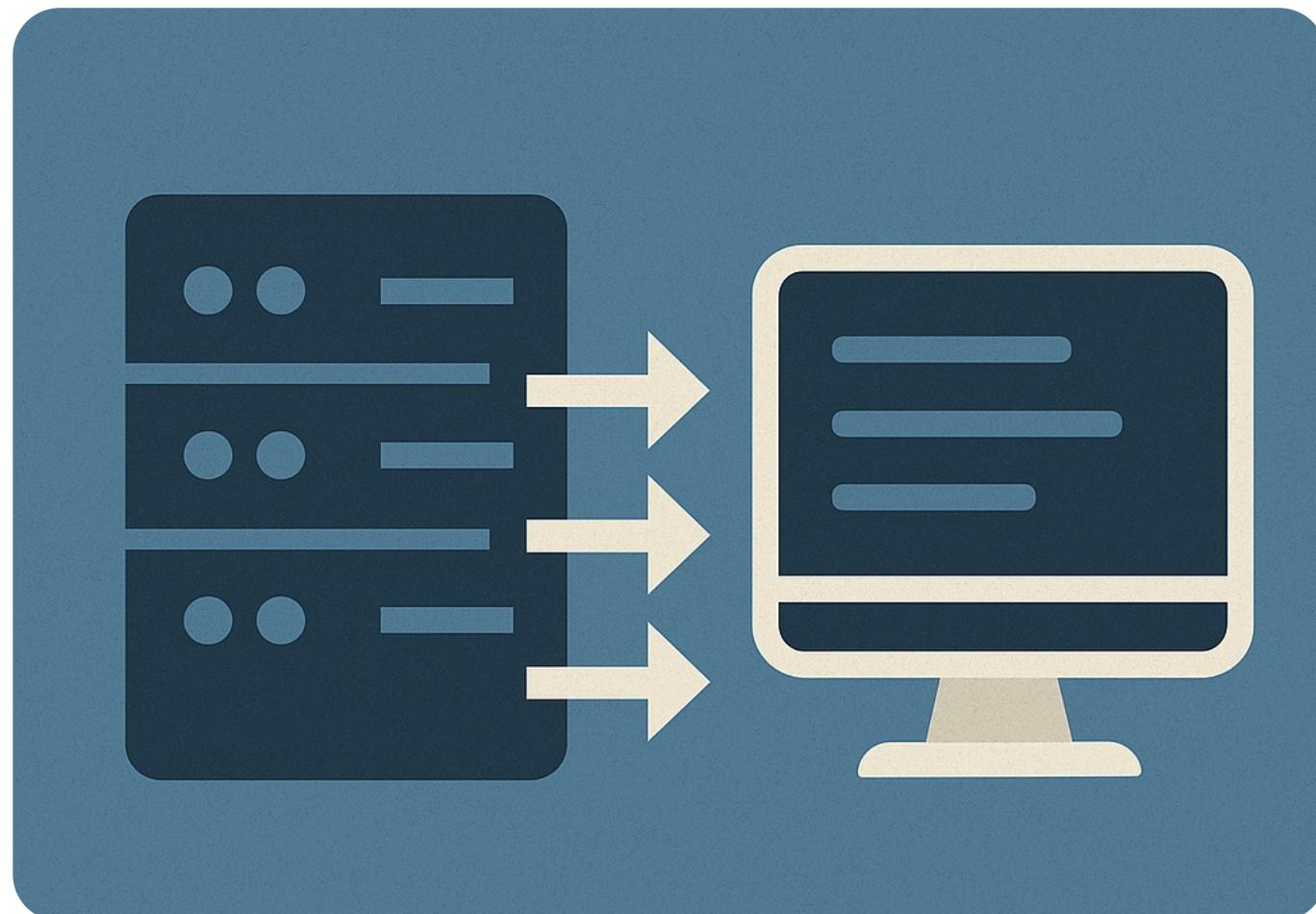
# User Experience

*Foco na experiência do*

**usuário**

**Quem nunca precisou ficar teclando F5 ou clicando em atualizar para esperar um dado ser cadastrado?** O usuário pode ter uma experiência muito melhor quando o Sistema notifica ele de atualizações. E notificações não são apenas uma badge em um ícone na toolbar.

# Server Sent Events

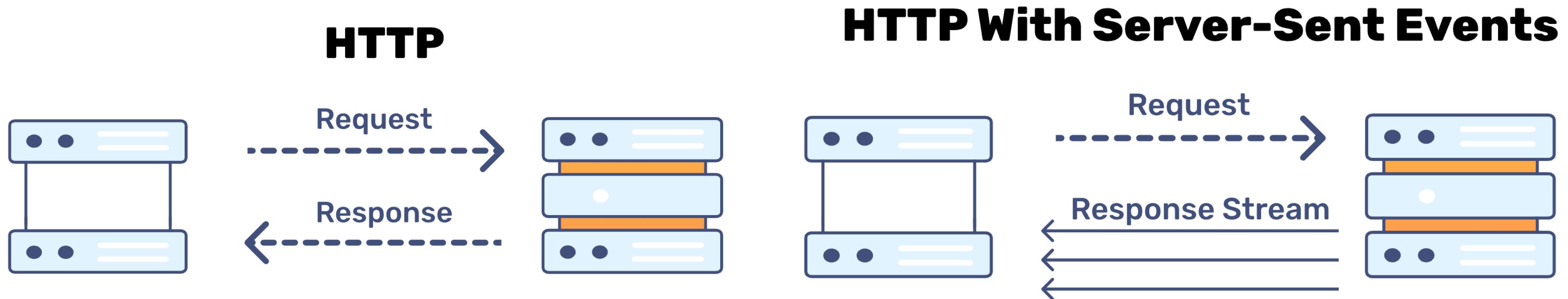


Server-Sent Events (SSE) é uma tecnologia de push baseada em HTTP que permite que servidores enviem atualizações automaticamente para clientes.

Diferente das requisições tradicionais em que o cliente é responsável por buscar dados do servidor, SSE inverte esse fluxo.

O suporte a SSE está presente em todos os navegadores modernos, como Firefox, Chrome, Safari, Opera e Microsoft Edge.

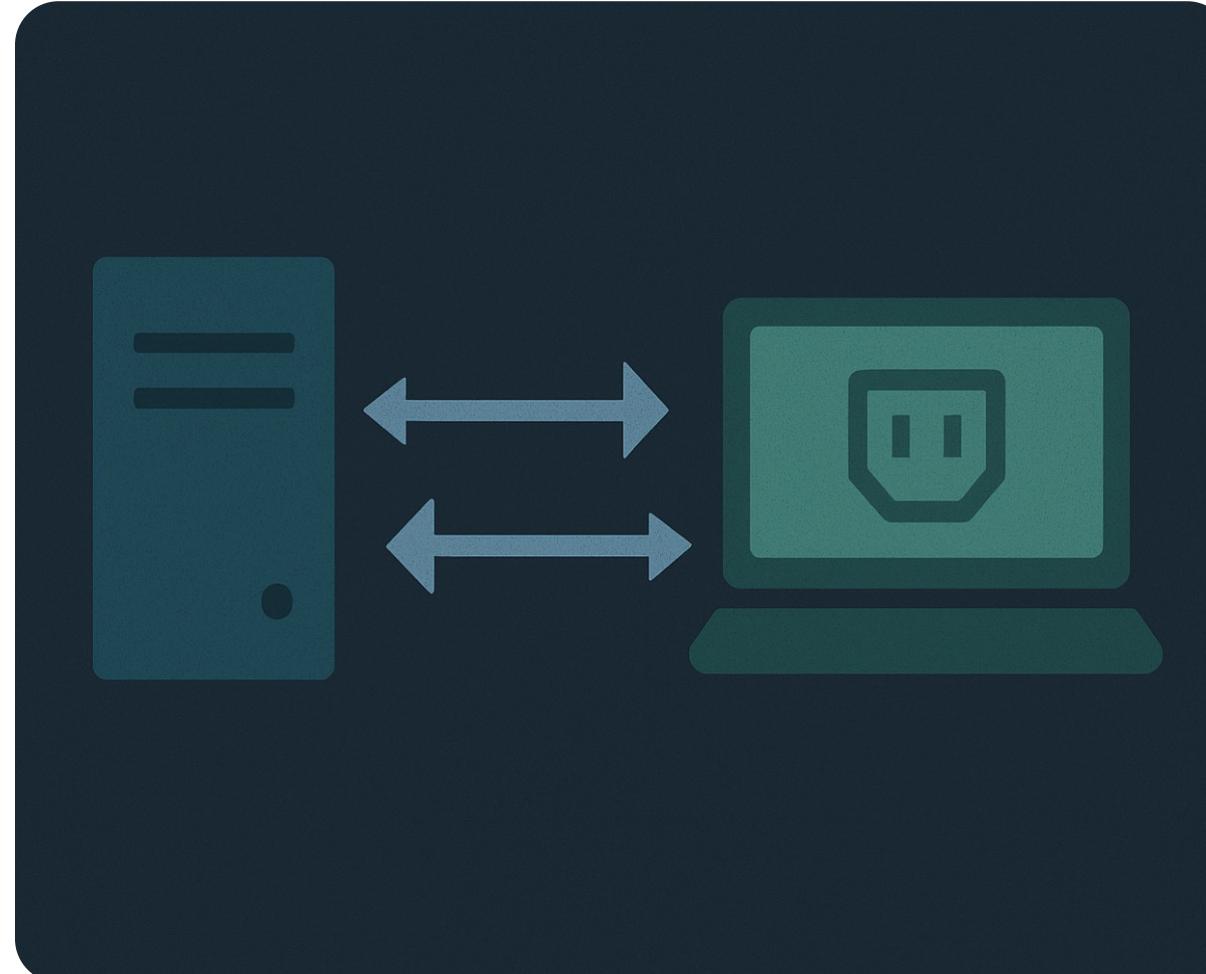
# Server Sent Events



# Demo SSR

```
isVideo = ( type == "Image" ) ||  
isUrl   = ( source.indexOf("youtu")  
            != -1 ) ||  
isElement = ( type == "URL" ) ||  
isObject  = ( type == "Element" ) ||  
            ( type == "Subject" )  
  
// Check if boxer is already active.  
if ($("#boxer").length > 1) {  
    return;  
}  
  
// Kill event  
_killEvent(e);  
  
// Cache internal data  
data = $.extend({}, {  
    $window: $(window),  
    $body: $("body"),  
    $target: $target,  
    $object: $object,  
    visible: false,  
    resizeTimer: null,  
    touchTimer: null,  
    library: {}  
});
```

# WebSockets

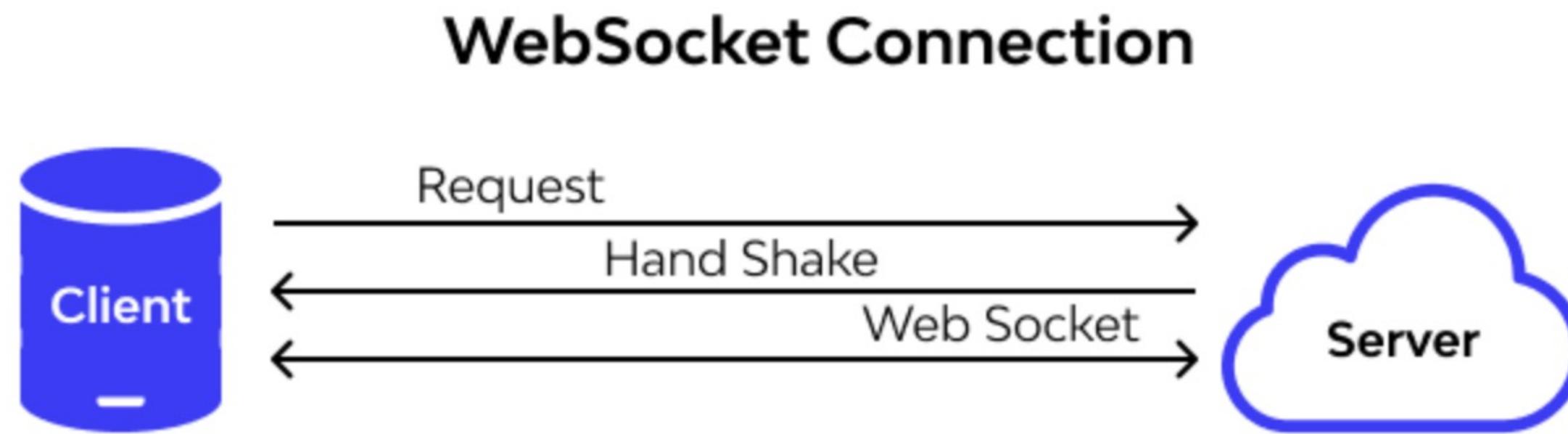


WebSocket é uma tecnologia de comunicação bidirecional que permite a transmissão de dados entre um navegador web e um servidor de maneira eficiente e em tempo real.

Diferente do modelo tradicional de requisições HTTP, onde o navegador faz uma solicitação e espera por uma resposta do servidor, com WebSockets, uma conexão persistente é mantida, permitindo a troca de mensagens de ambos os lados a qualquer momento.

Apesar de semelhante ao SSR, WebSockets permite a troca de informações de forma bi-direcional.

# WebSockets



# SignalR



SignalR é uma tecnologia orientada a eventos que habilita recursos em tempo real em aplicações .NET gerenciando conexões persistentes entre o servidor e os clientes conectados.

Ele negocia o transporte mais eficiente disponível, priorizando WebSockets e recorrendo a Eventos Enviados pelo Servidor (SSE) ou a long-polling quando necessário.

Microsoft Azure prove um serviço nativo de SignalR de modo que você não precisa implementar o Hub manualmente e permite maior controle e escalabilidade em cima do Hub.

# Demo SignalR

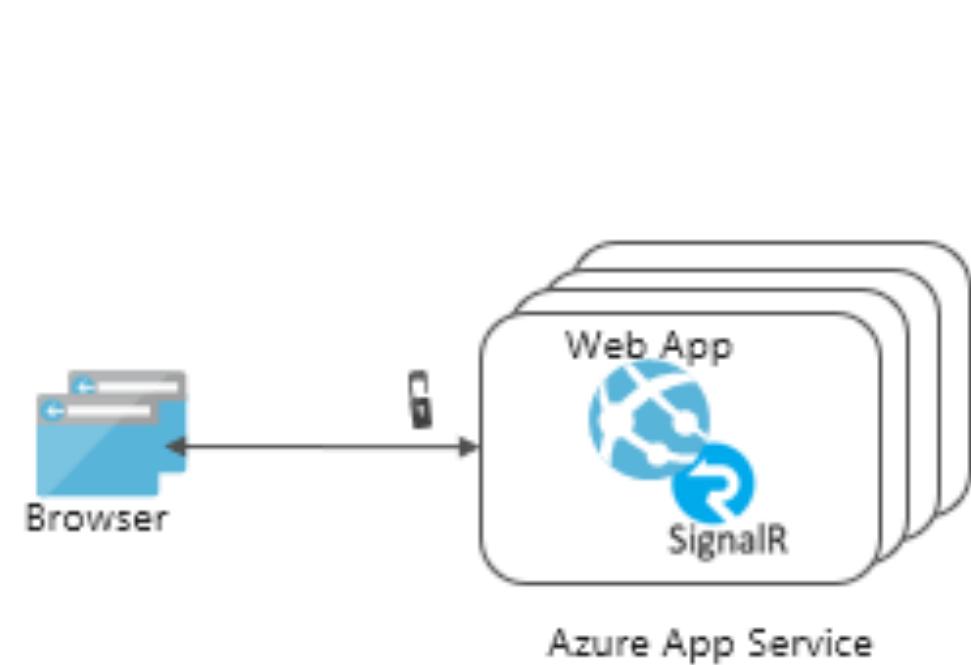
```
isVideo = ( type == "Image" ) ||  
isUrl   = ( source.indexOf("youtu")  
            != -1 ) ||  
isElement = ( type == "URL" ) ||  
isObject  = ( type == "Element" ) ||  
            ( type == "Subject" )  
  
// Check if boxer is already active.  
if ($("#boxer").length > 1) {  
    return;  
}  
  
// Kill event  
_killEvent(e);  
  
// Cache internal data  
data = $.extend({}, {  
    $window: $(window),  
    $body: $("body"),  
    $target: $target,  
    $object: $object,  
    visible: false,  
    resizeTimer: null,  
    touchTimer: null,  
    library: {}  
});
```

# Azure SignalR

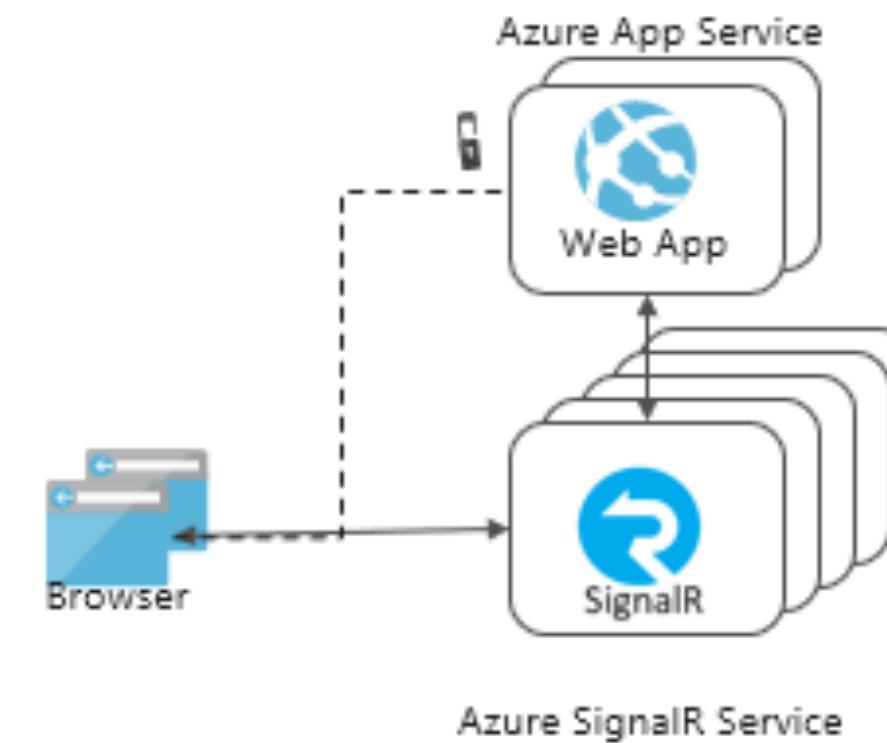


O Azure SignalR Service simplifica o processo de adição da funcionalidade Web em tempo real aos aplicativos por HTTP. Essa funcionalidade em tempo real permite que o serviço envie por push atualizações de conteúdo a clientes conectados, como um aplicativo móvel ou Web de página única. Como resultado, os clientes são atualizados sem a necessidade de sondar o servidor, ou enviar novas solicitações HTTP para atualizações.

# Azure SignalR



Self-hosted SignalR

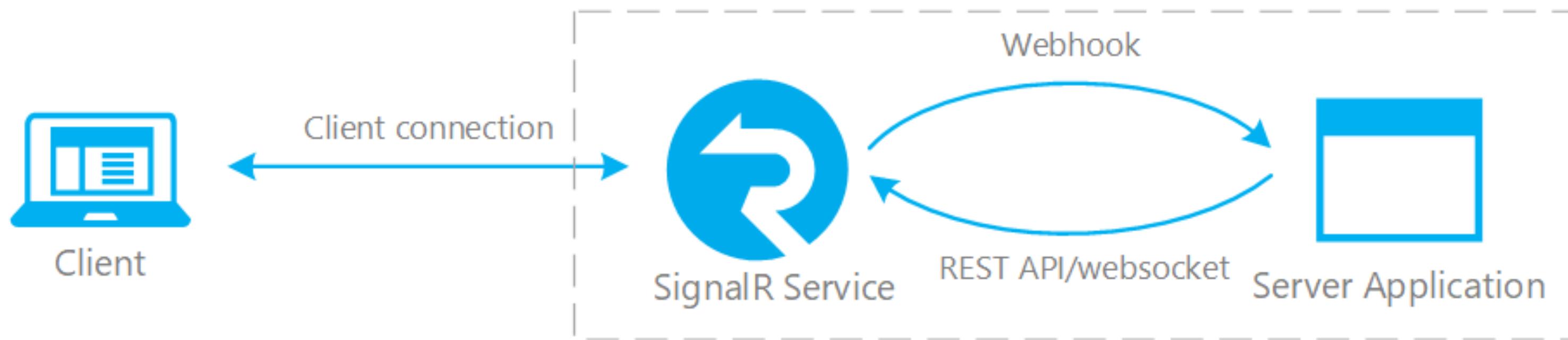


Managed SignalR Service

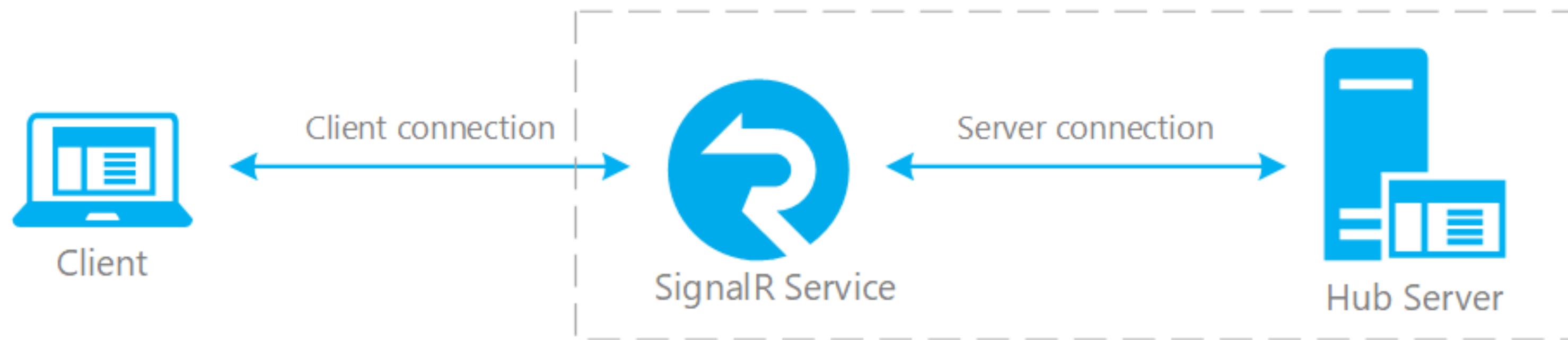
# Azure SignalR

	Gratuito	Padrão	Premium
Conexões simultâneas por unidade	20	1.000	1.000
Mensagens/Unidade/Dia	20.000	ilimitado (gratuito para o primeiro milhão de mensagens)	ilimitado (gratuito para o primeiro milhão de mensagens)
Preço/Unidade/Dia	Gratuito	<b>\$1,61</b>	<b>\$2</b> (escala de instância <= 1,000 unidades)
SLA	N/A	99,9%	99,95%
Suporte à Zona de Disponibilidade	-	-	Sim
Escalonamento automático totalmente gerenciado	-	-	Sim
Nome de domínio personalizado	-	-	Sim
Número máx. de unidades	1	100	1.000
Replicação geográfica	-	-	Sim

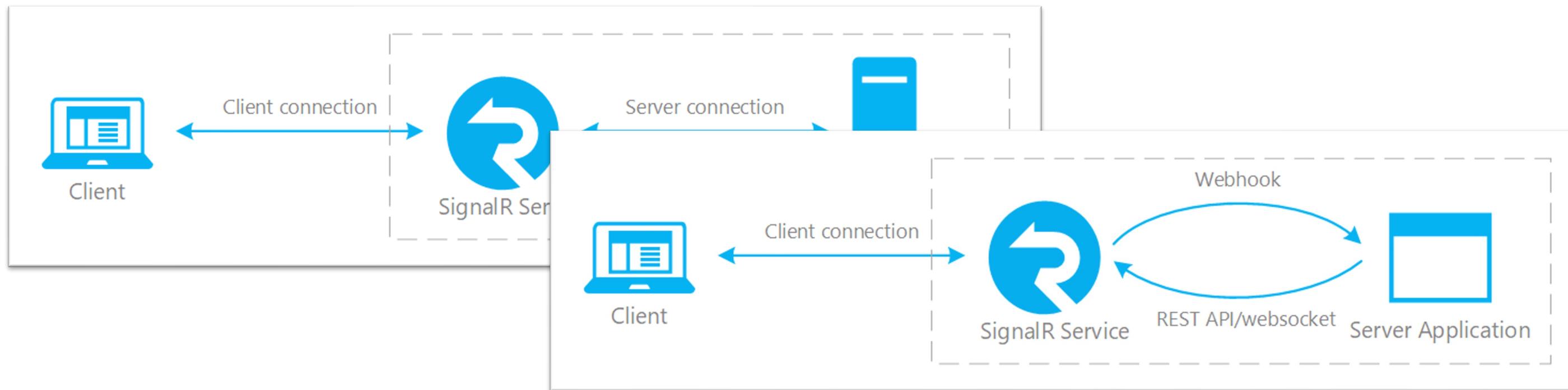
# Modo Padrão



# Modo Serverless



# Modo Clássico



O modo Clássico é principalmente para compatibilidade com versões anteriores para os aplicativos criados antes de existir um modo Padrão e Sem servidor. Não use o modo Clássico, exceto como último recurso. Para novos aplicativos, escolha Padrão ou Sem servidor com base em seu cenário. Você deve considerar reprojetar aplicativos existentes para eliminar a necessidade do modo Clássico.

# Demo SignalR

```
isVideo = ( type == "Image" ) ||  
isUrl   = ( source.indexOf("youtu")  
            != -1 ) ||  
isElement = ( type == "URL" ) ||  
isObject  = ( type == "Element" ) ||  
            ( type == "Subject" )  
  
// Check if boxer is already active.  
if ($("#boxer").length > 1) {  
    return;  
}  
  
// Kill event  
_killEvent(e);  
  
// Cache internal data  
data = $.extend({}, {  
    $window: $(window),  
    $body: $("body"),  
    $target: $target,  
    $object: $object,  
    visible: false,  
    resizeTimer: null,  
    touchTimer: null,  
    library: {}  
});
```

# Demos e Slides

[https://github.com/gustavobigardi  
/mvpconf-2025-blazor-ssr-signalr](https://github.com/gustavobigardi/mvpconf-2025-blazor-ssr-signalr)



# Obrigado!





Edição 2025

**Fale com a gente e  
descubra como fazer  
parte dessa transformação.**

✉ contato@mvpconf.com.br

✉ www.mvpconf.com.br

○ / @mvpconf

