PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IIC3103

Taller de Integración

Profesores

Arturo Tagle / Daniel Darritchon

Resumen clase anterior

REST

Es un estándar de desarrollo de servicios web.

Se definen reglas y protocolos para el diseño de servicios, pero no imponen un estándar formal.

Características de servicios REST

1. Arquitectura cliente servidor
2. Sistema por capas
3. Caché
4. Servidor stateless
5. Interfaz uniforme

Resumen clase anterior

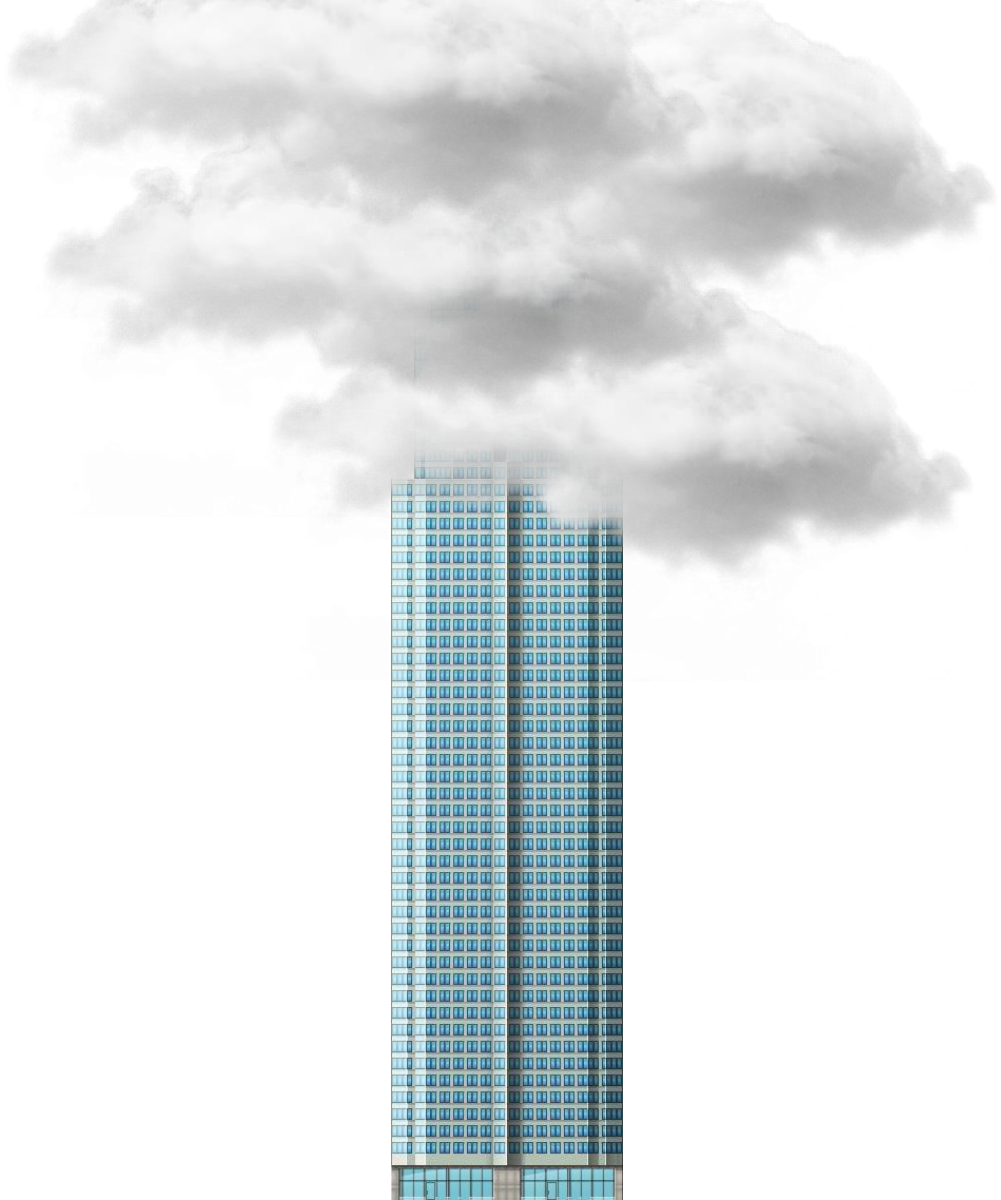
Seguridad en servicios

Autenticación: ¿Eres quién dices ser?

Autorización: ¿Estás autorizado para realizar esta acción?

Dudas Tarea 1

¿Ya partieron la tarea?



Integración por eventos

Módulo 3 - Parte 1

Introducción a eventos

Motivación

Queremos saber todo lo que pasa, cuando pasa.

Queremos saber cualquier cambio de estado de un sistema, un objeto, una acción, etc.

No queremos perder ningún dato que podría representar un cambio en una tendencia, evento importante, nuevos comportamientos, etc.

LarrainVial

INDICADORES DE MERCADO

Chile

Índices	Puntos	Var.%
IPSA	4.898,69	-1,42
IGPA	25.042,59	-1,66
S&P 500	4.431,03	-1,28
S&P 100	2.025,72	-1,50
NASDAQ 100	14.051,30	-1,93
NASDAQ COMP	13.465,00	-1,79
S&P 40 MILA	553,69	-0,71
BOVESPA	117.319,00	-0,85
MERVAL	90.828,61	-1,37
COLCAP	1.625,26	-0,30
SP/BVL PERU SEL	637,20	-0,24
FTSE 100	*7.618,31*	*-0,67*
DAX	*14.192,70*	*-0,64*
IBEX 35	*8.585,00*	*-0,25*
NIKKEI 225	*26.821,52*	*-0,61*
HANG SENG	*21.208,30*	*-3,03*
CSI 300	*4.100,07*	*-3,09*

IPSA ☆ Mercado Local Abierto

Buscar.. Buscar Comparar

Último 4.898,69 | Var. Pts. -70,56 | Var. % -1,42

1D 5D 1M 3M 1A 3A | Desde | Hasta

Última actualización: 11/04/2022 14:45:26

Principales Acciones *	Precio	Var. %
AESANDES	135,14	0,00
AGUAS-A	165,31	-2,76
ANDINA-B	1.783,10	-1,44
BCI	28.906,00	-1,00
BSANTANDER	43,35	+0,70
CAP	12.616,00	-1,82
CCU	5.773,30	-2,15
CENCOSUD	1.518,90	-0,07
CHILE	85,20	+0,24
CMPC	1.400,00	-0,33

Mayores Alzas	Precio	Var.%
EWCCL	32.809,00	+76,19
IYKCL	171.170,00	+14,48
WFCCL	40.211,00	+6,66
PFECL	44.382,00	+6,61
CALICHERAA	600,00	+6,24
IYRCL	88.365,00	+3,17
IGE CL	33.392,00	+3,17
ORCLCL	66.503,00	+3,11
HONCL	154.340,00	+2,04
EEMCL	36.397,00	+1,53

Mayores Bajas	Precio	Var.%
NKECL	104.080,00	-27,37
EWHCL	18.220,00	-7,88
CGEGAS	192,07	-6,16
LTM	221,60	-4,97
PUCOBRE	4.418,00	-4,82
SQM-B	66.610,00	-4,64
MSFTCL	234.000,00	-4,56
LQDCL	94.705,00	-4,53
ENELGXCH	127,22	-4,29
ECL	426,25	-4,21

Más transadas	MM ($)	Var.%
SQM-B	31.378,04	-4,64
FALABELLA	6.477,41	-0,67
VAPORES	5.535,14	-3,27
ENELAM	4.400,66	-0,29
CHILE	2.317,26	+0,24
CAP	2.156,96	-1,82
COPEC	2.101,40	-1,28
BSANTANDER	1.608,54	+0,70
ITAUCORP	1.119,88	-0,98
CMPC	1.046,58	-0,33

¿Sería eficiente un servicio que esté preguntando por cada cambio?

Indicadores	Valor	Var.%
UF	31.795,21	+0,06
UTM	55.704,00	+0,30
Dólar Observado	814,28	+0,79
IPC Marzo	118,26	+1,90

Commodities	Valor (USD)	Var.%
COBRE A CASH	4,6427	-1,49
PETRÓLEO WTI	94,2300	-2,59
ORO OZ.	1.948,03	+0,17
PLATA 5000	24,85	+0,21
CELULOSA NBSK	1.205,04	0,00
HARINA PESCADO	1.442,75	-1,27
GAS NATURAL	4,3710	+4,20

Renta Fija	Tasa	Var.
BCU 5 AÑOS (Delay 48 hrs.)	1,69	+0,07
BCU 10 AÑOS (Delay 48 hrs.)	2,00	0,00
BCU 20 AÑOS (Delay 48 hrs.)	2,30	0,00
BCP 2 AÑOS (Delay 48 hrs.)	5,70	+0,05
BCP 5 AÑOS (Delay 48 hrs.)	6,64	+0,08
BCP 10 AÑOS (Delay 48 hrs.)	6,34	+0,05
DEP $ 3 MES	0,24	-0,06
DEP $ 1 AÑO	8,32	+0,20
DEP UF 1 AÑO	0,44	-0,34

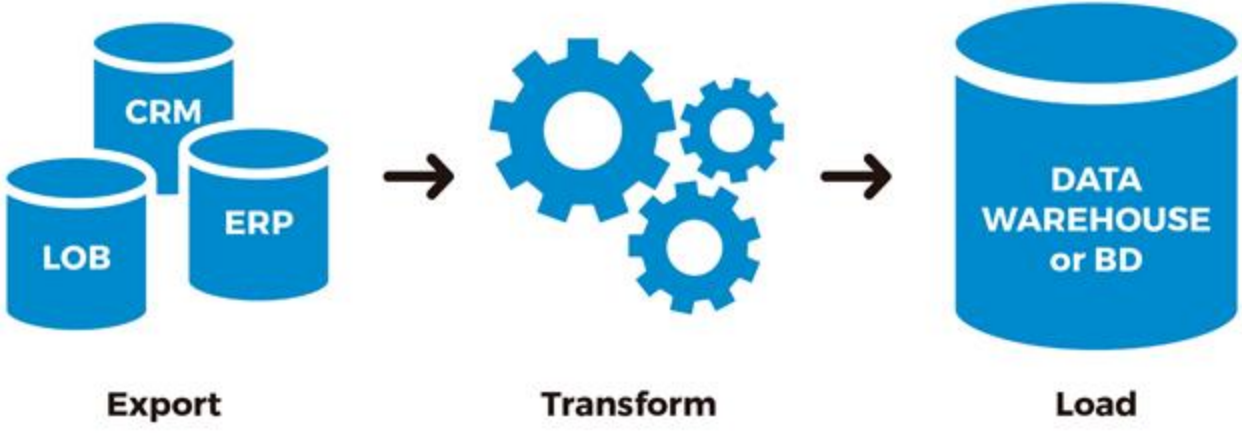
Monedas	Valor	Var.%
DÓLAR (USD/CLP)	815,9000	+0,17
EURO (EUR/USD)	1,0890	+0,12
REAL (USD/BRL)	4,6959	-0,69
PESO MEX. (USD/MXN)	20,1772	+1,14
DÓLAR AUS. (AUD/USD)	0,7422	-0,44
YEN (USD/JPY)	125,5290	+0,94
LIBRA (GBP/USD)	1,3027	+0,09
PESO COL. (USD/COP)	3.742,67	-0,58
SOL PER. (USD/PEN)	3,7150	+0,01

Motivación

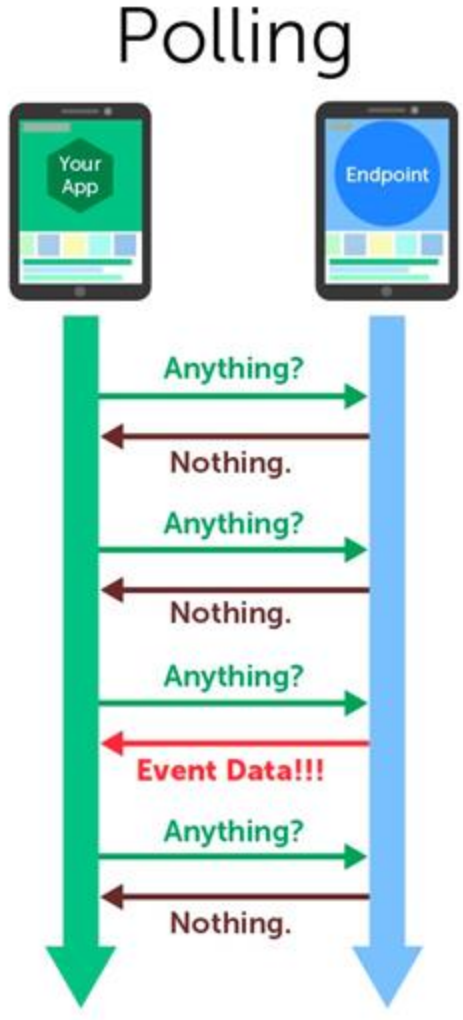
Más ejemplos reales:

- Sincronización de datos.
- Evitar pollings
- Mantener registros históricos / Auditoría.
- Viajar en el tiempo.

Sincronización de datos por ETL

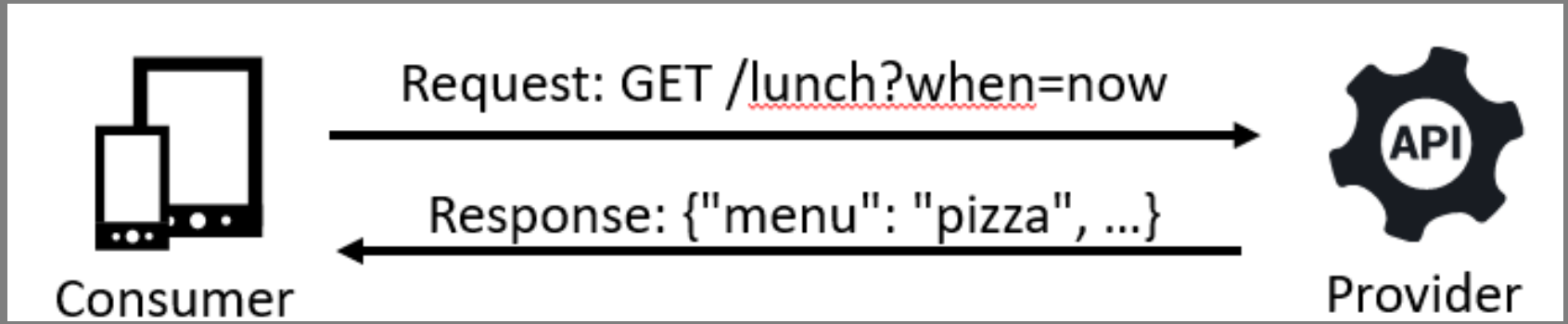


Evitar polling

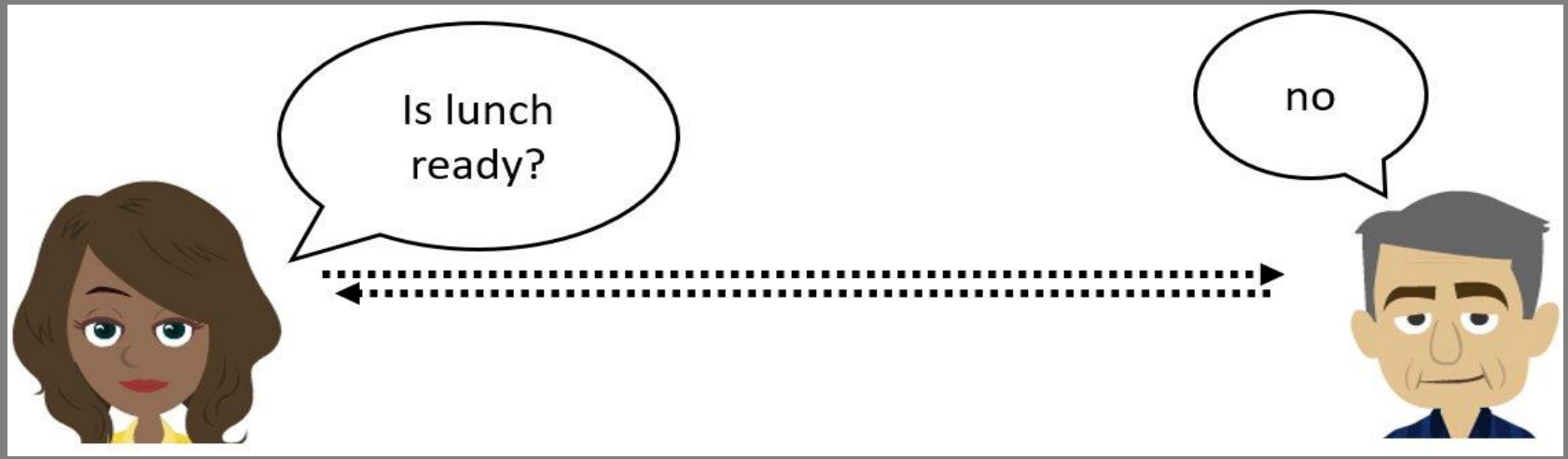


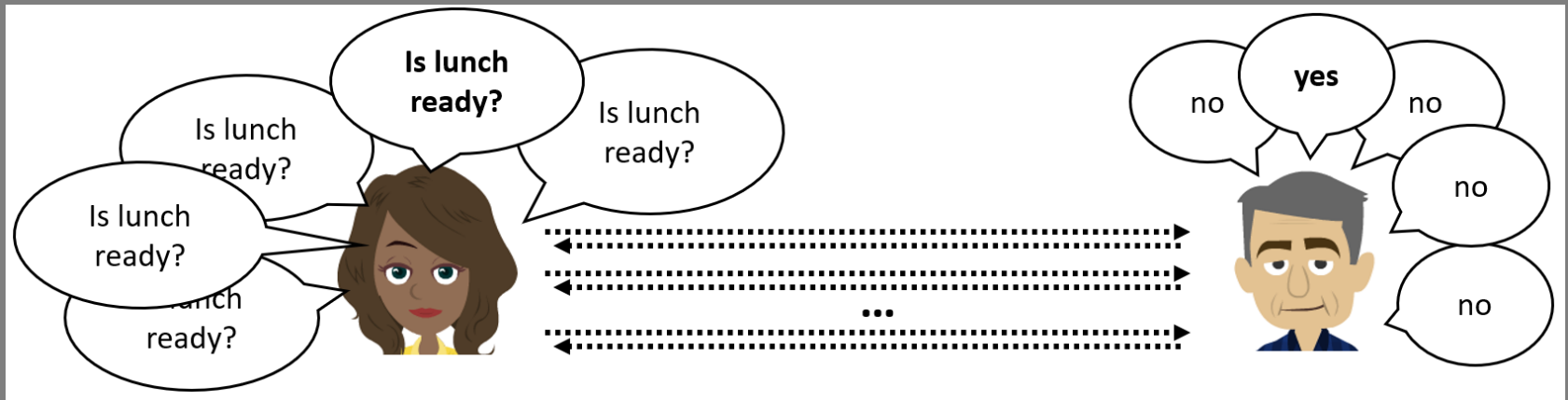
Más ejemplos

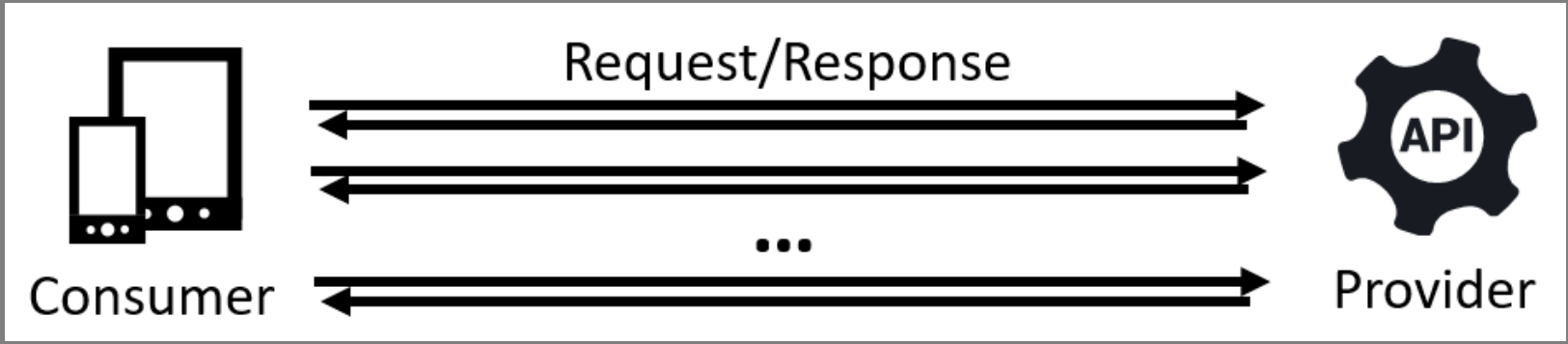
Analogía de interacción humana



https://blog.axway.com/learning-center/apis/basics/event-driven-vs-rest-api-interactions

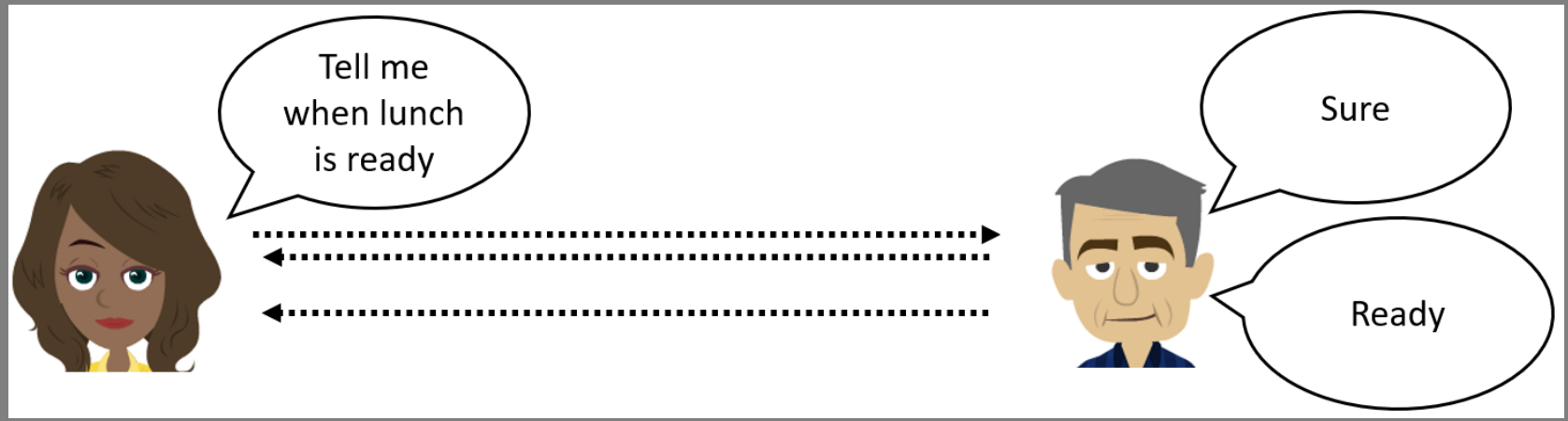

Is lunch ready?
no

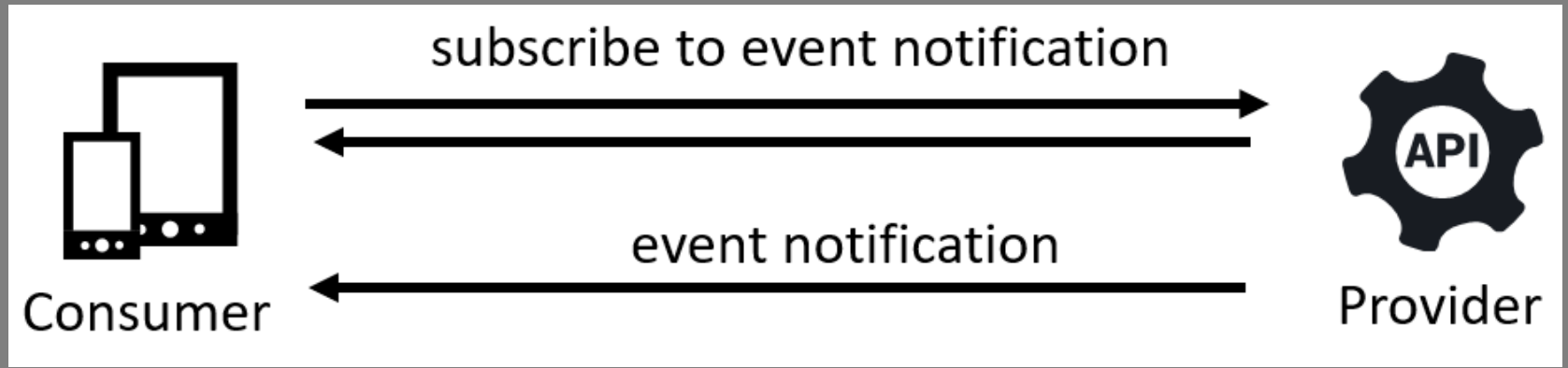

Is lunch ready?
Is lunch ready?
Is lunch ready?
Is lunch ready?
Is lunch ready?
no
yes
no
no
no



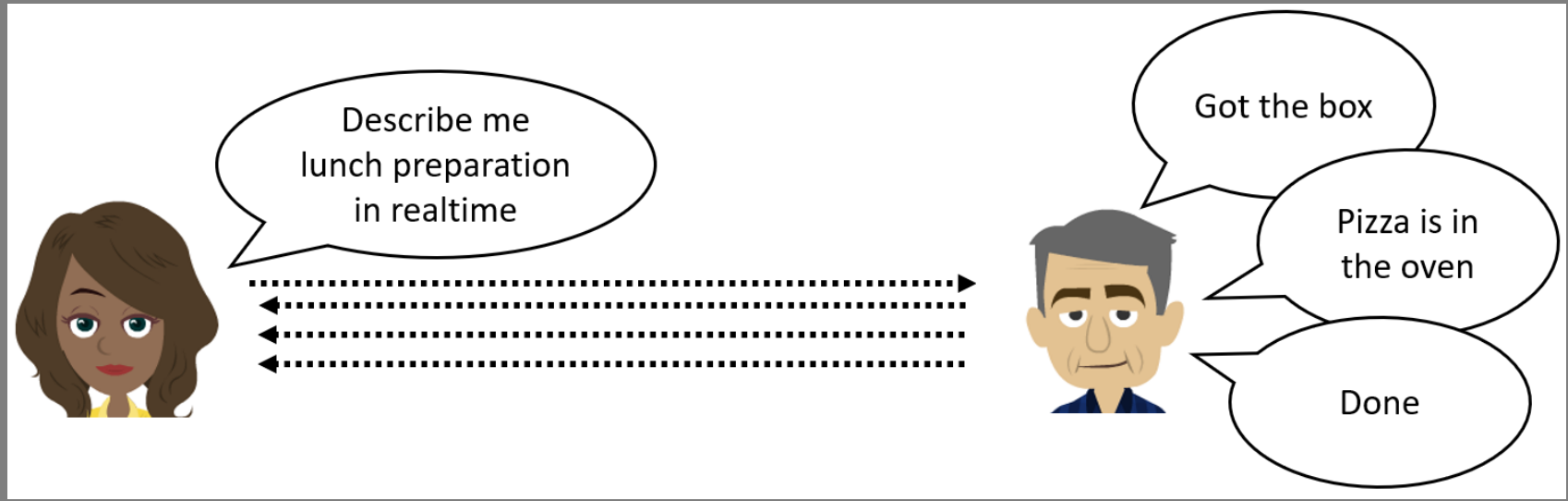
Computers are not annoyed, but it’s much more a development effort on the consumer side and a waste of resources on the provider side, often more than they can provide. **It doesn’t seem like the “right” solution**.

Webhook


Tell me
when lunch
is ready
Sure
Ready

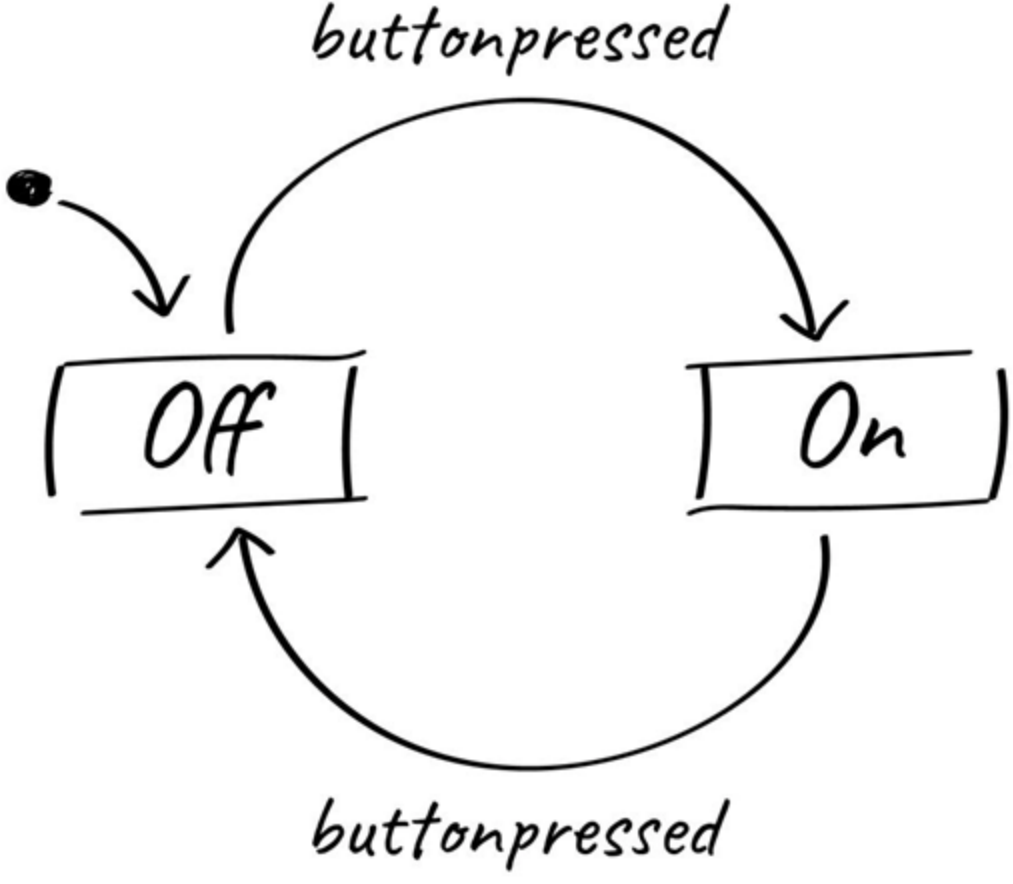

subscribe to event notification
event notification
API
Consumer
Provider

Streaming


Describe me
lunch preparation
in realtime
Got the box
Pizza is in
the oven
Done

Algunos conceptos

- Máquina de estados
- Estado
- Transición
- Evento


buttonpressed
Off
On
buttonpressed

En general, podemos encontrar las siguientes ventajas en una arquitectura basada en eventos

1. Menos consultas al sistema.
    a. Menor overhead y menor carga => mayor escalabilidad
2. Mejor monitoreo.
3. Hechos son transmitidos instantáneamente, o cercano a tiempo real.

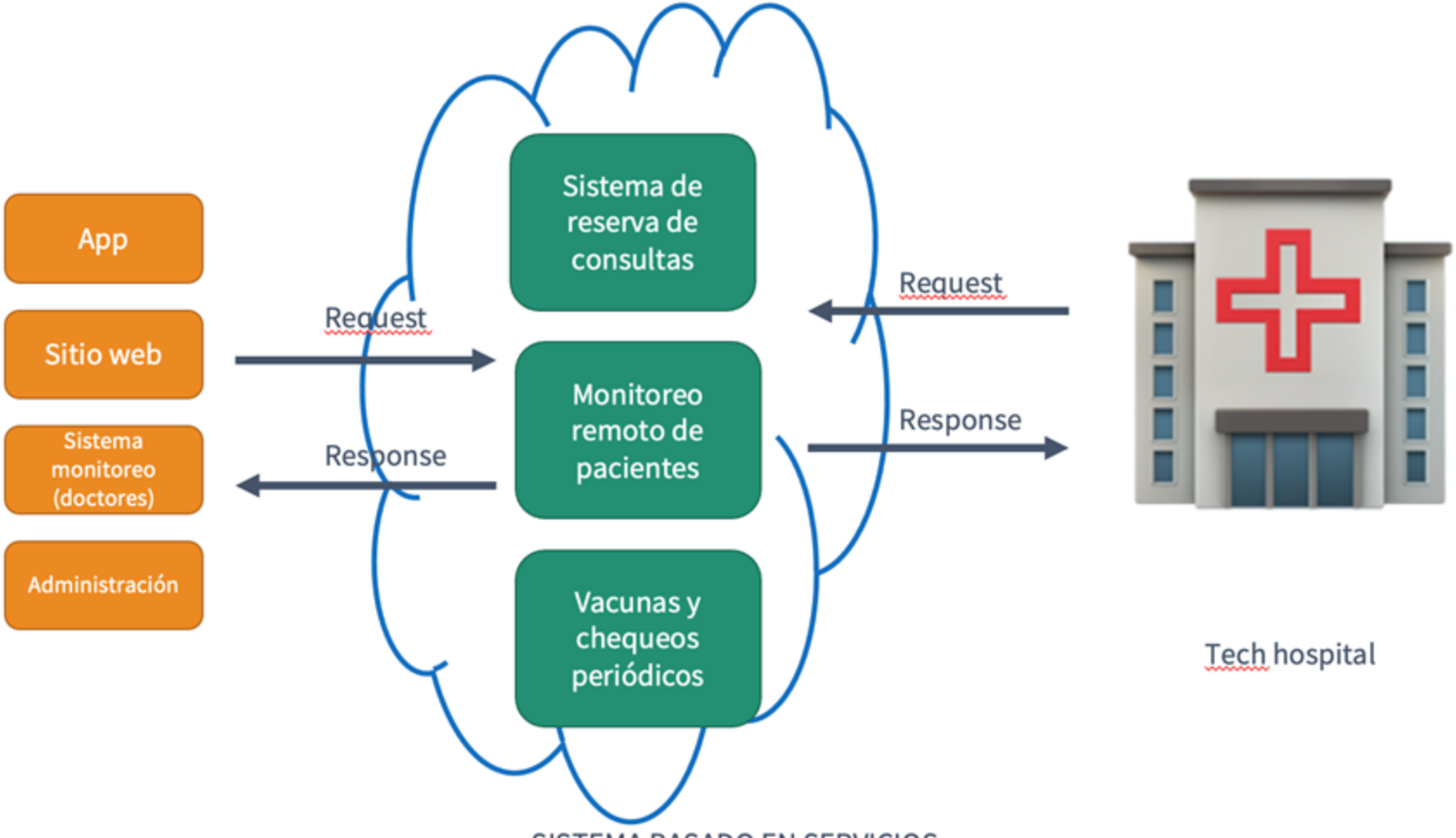
Algunos sistemas se ven beneficiados con sistemas pensados en eventos en vez de servicios

Aplicando conceptos

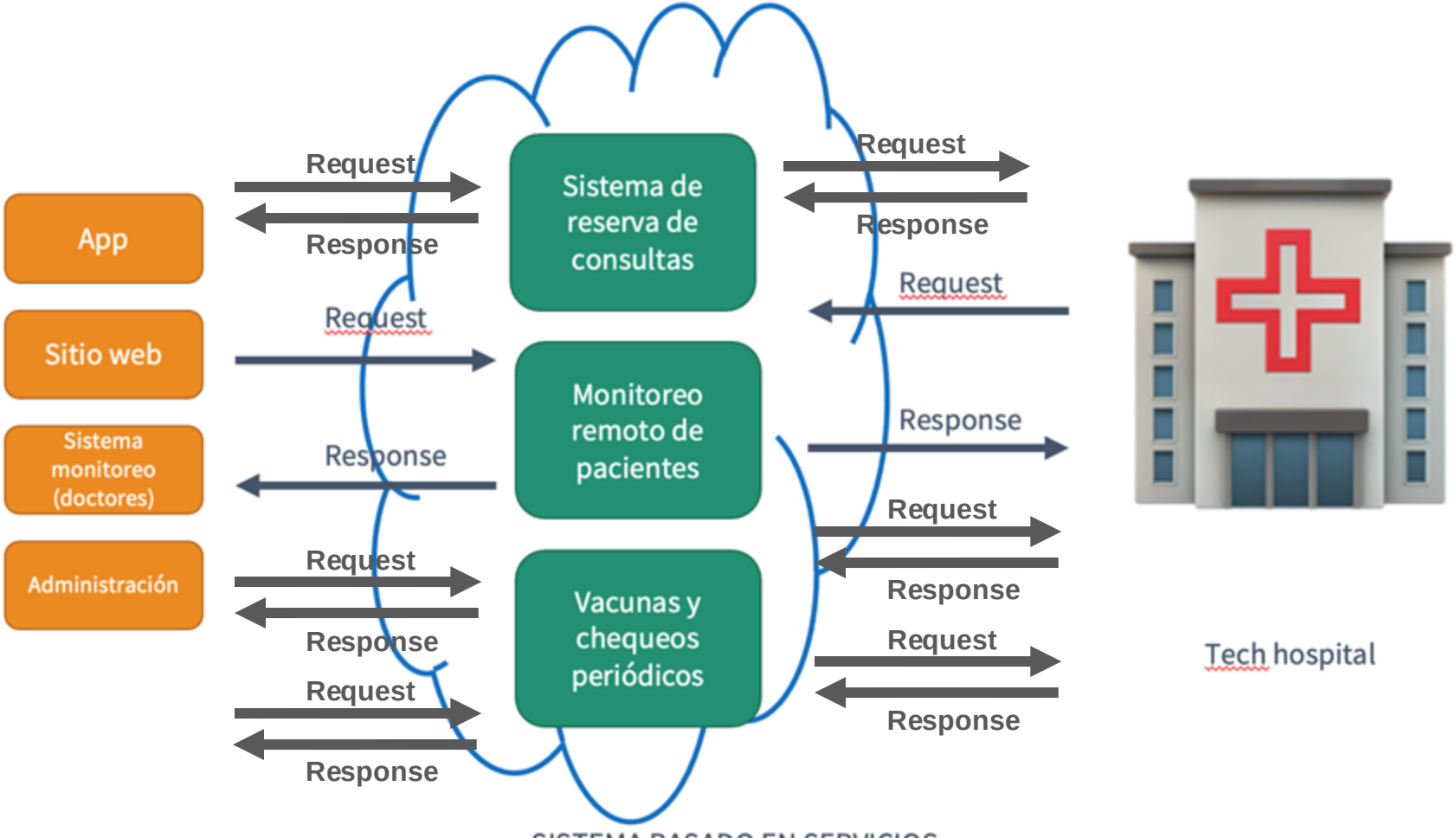
- **Evento:** Hecho que sucede en un software o hardware que produce un cambio de estado.
- **Emisor:** Agente que genera el evento
- **Suscriptores:** Todos los clientes que reciben el evento
- **Notificación:** Acción de propagar el evento desde el emisor a los suscriptores.

Aplicando los ejemplos anteriores

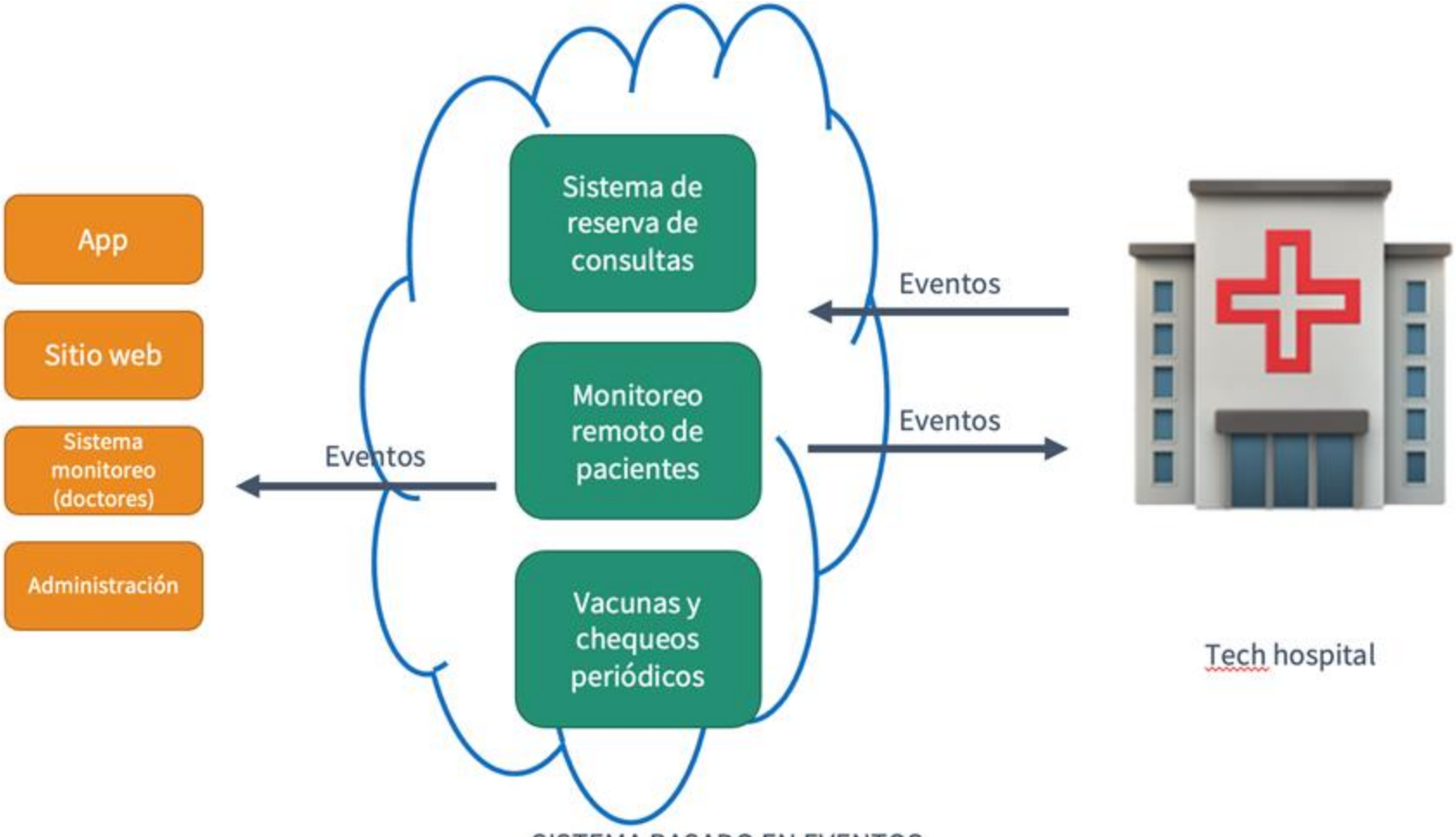
Tech hospital



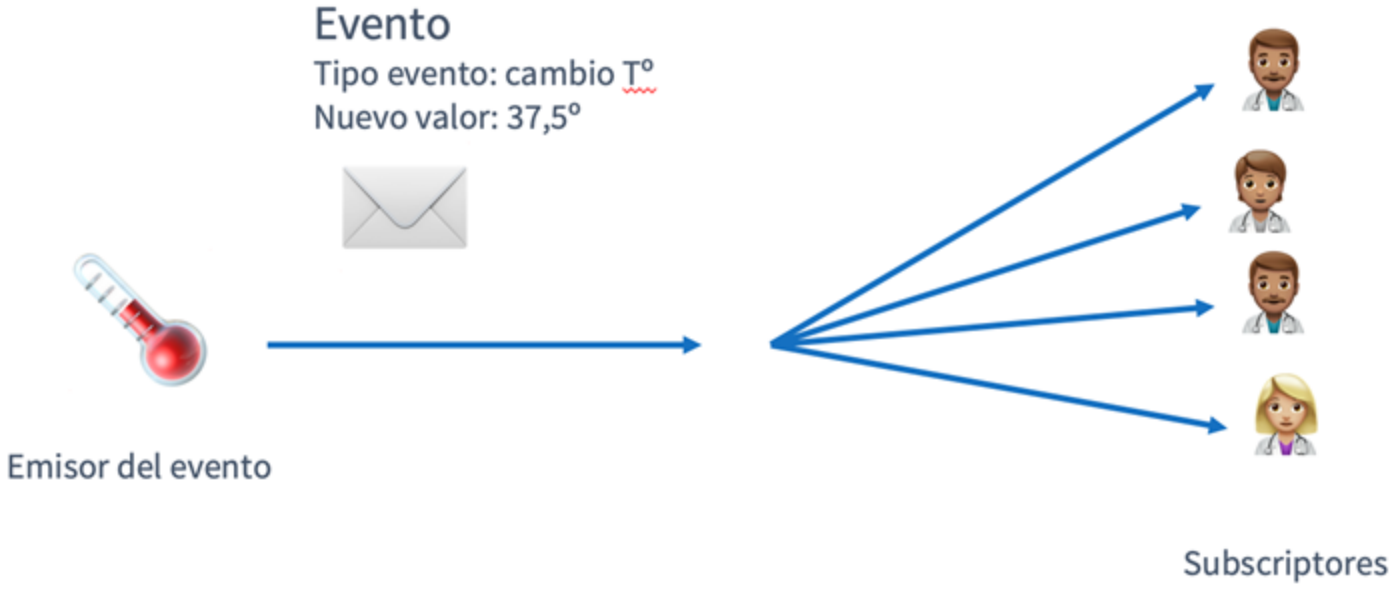
SISTEMA BASADO EN SERVICIOS

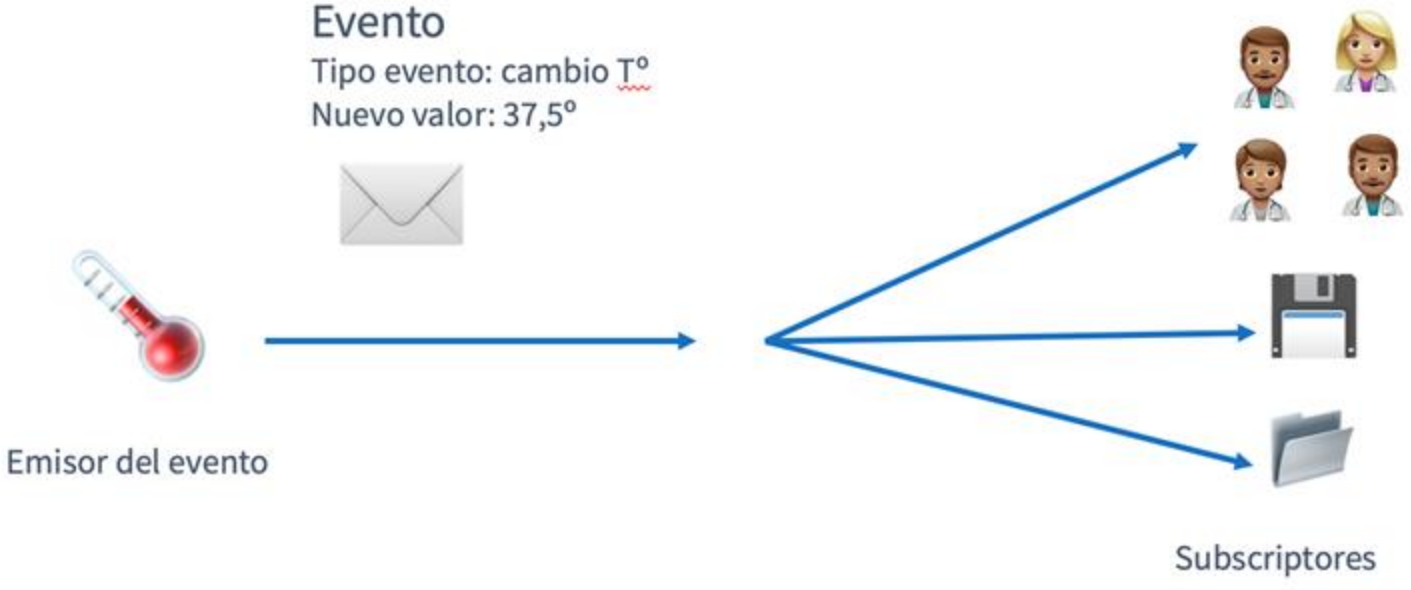


SISTEMA BASADO EN SERVICIOS



SISTEMA BASADO EN EVENTOS


Evento
Tipo evento: cambio T°
Nuevo valor: 37,5°
Emisor del evento
Subscriptores


Evento
Tipo evento: cambio T°
Nuevo valor: 37,5°
Emisor del evento
Subscriptores

Características de los eventos

1. Bajo acoplamiento.

2. La comunicación es entre múltiples sistemas.

3. La comunicación es iniciada por el emisor del evento.

4. La comunicación es asíncrona.

5. La comunicación sólo tiene un mensaje: el evento.

6. Suscriptores deben registrarse para recibir eventos.

1. Bajo acoplamiento

Permite hacer modificaciones del funcionamiento interno del sistema que gatilla el evento:

- Los sistemas suscriptores no se ven impactados.
- El cuerpo del evento y medio de transmisión no cambia.

Permite hacer modificaciones en el consumidor del servicio

- Se puede cambiar al consumidor del evento sin problemas para el emisor u otros consumidores.

1. Bajo acoplamiento

Desacoplamiento tecnológico

- El evento publicado puede comunicarse con un suscriptor con cualquier tecnología.
- El sistema que emite el evento es “caja negra” para el suscriptor. El suscriptor sólo espera una evento según el contrato (documentación).

2. La comunicación es entre múltiples sistemas

Existen tres actores:

- Emisor del evento: quién genera el cambio de estado
- Concentrador o bus de eventos (opcional): Sistema encargado de gestionar suscripciones y propagación del evento.
    - Apache Kafka
    - RabbitMQ
    - GCP Pubsub
    - AWS Event Bridge
- Suscriptor: quién recibe el evento

3. La comunicación es iniciada por el emisor del evento.

La comunicación se inicia cuando el emisor detecta un cambio de estado que debe ser notifcado.

La comunicación termina cuando el mensaje llega a el/los suscriptores.

El suscriptor **nunca** inicia la comunicación, salvo por las acciones necesarias para suscribirse a los eventos.

4. La comunicación es asíncrona.

El emisor envía un evento y no espera una respuesta del suscriptor.

El emisor opera bajo una lógica de “fire & forget”.

5. La comunicación sólo tiene un mensaje: el evento.

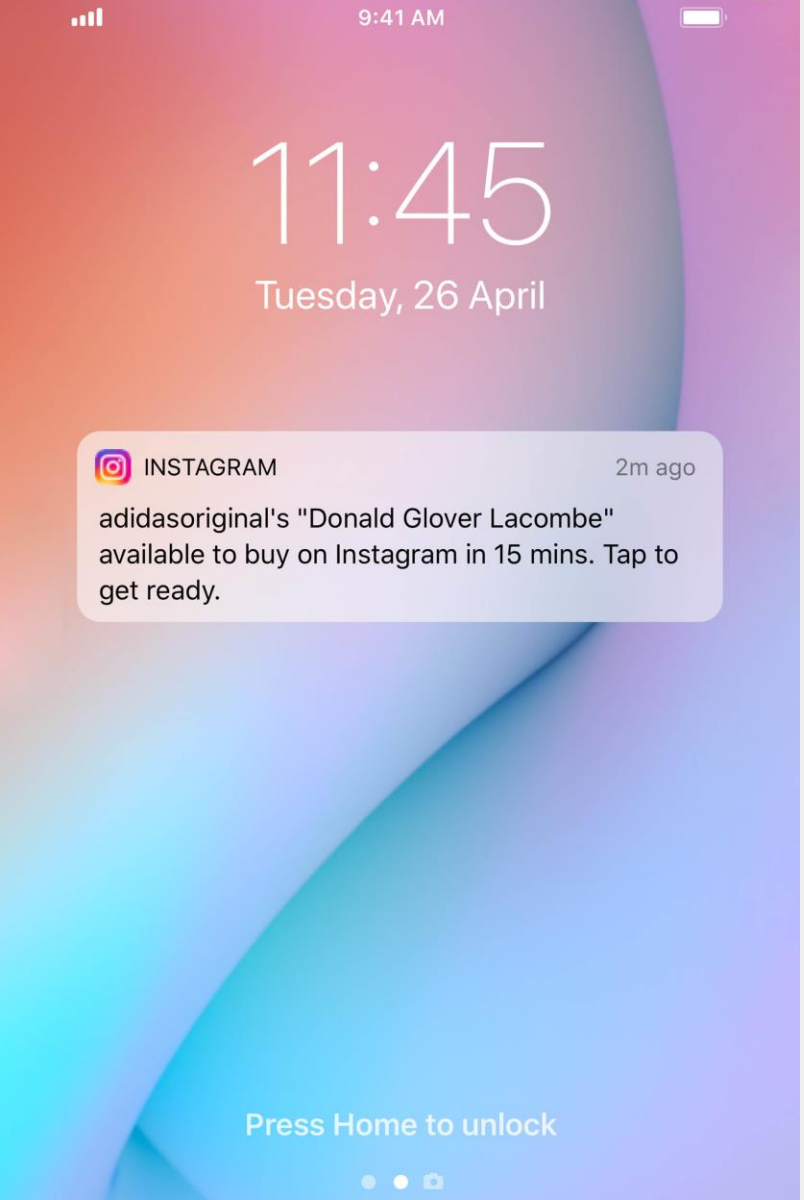
El emisor envía un mensaje con toda la información relevante para el evento.

No existen más comunicaciones entre las partes.

6. Suscriptores deben registrarse para recibir eventos.

La comunicación sólo es posible cuando el suscriptor se ha registrado para recibir eventos.

Sin la acción de la suscripción, el emisor no puede conocer a los receptores de los eventos.



Ejemplo

Notificación de redes sociales

Flujo para recibir notificaciones

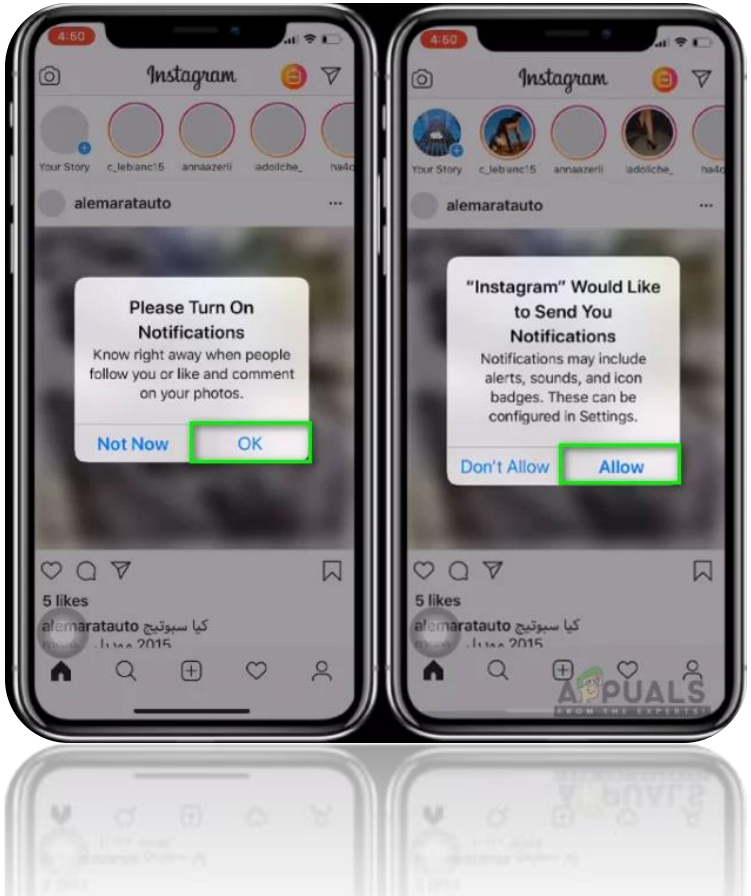
1. Suscriptor se registra para recibir notificaciones. En el caso de una app:
    - Descargar e instalar aplicación.
    - Aceptar recibir notificaciones o autorizar permiso en menú del sistema operativo.
2. Se genera un token de notificación en el **servidor de notificaciones (OSNPS)**, la cual es utilizada por la red social para enviar la notificación.

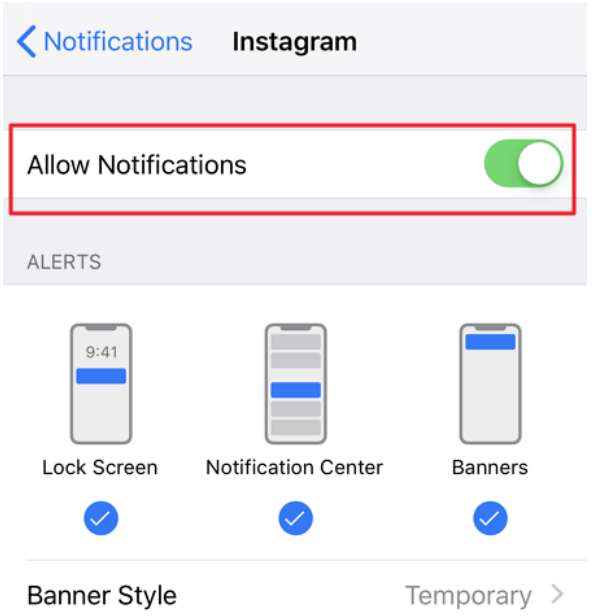
Servidor de notificaciones (OSPNS)

Apple Push Notification service 

Firebase Cloud Messaging 

Autorizar notificaciones





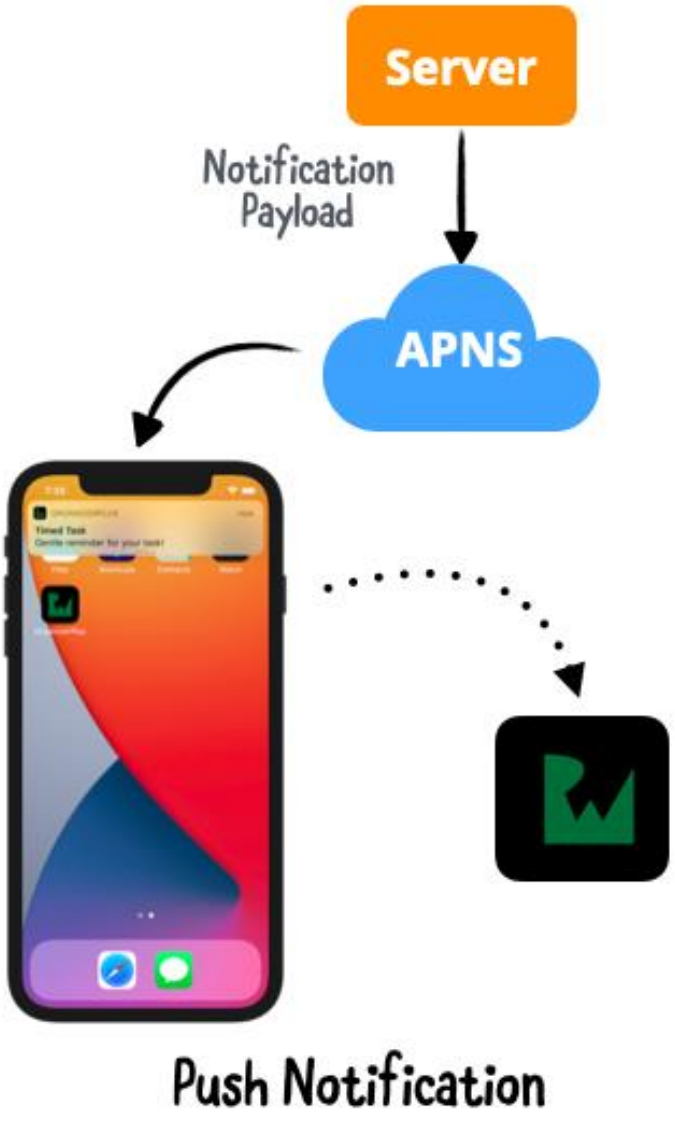
Esperar a recibir una nueva notificación

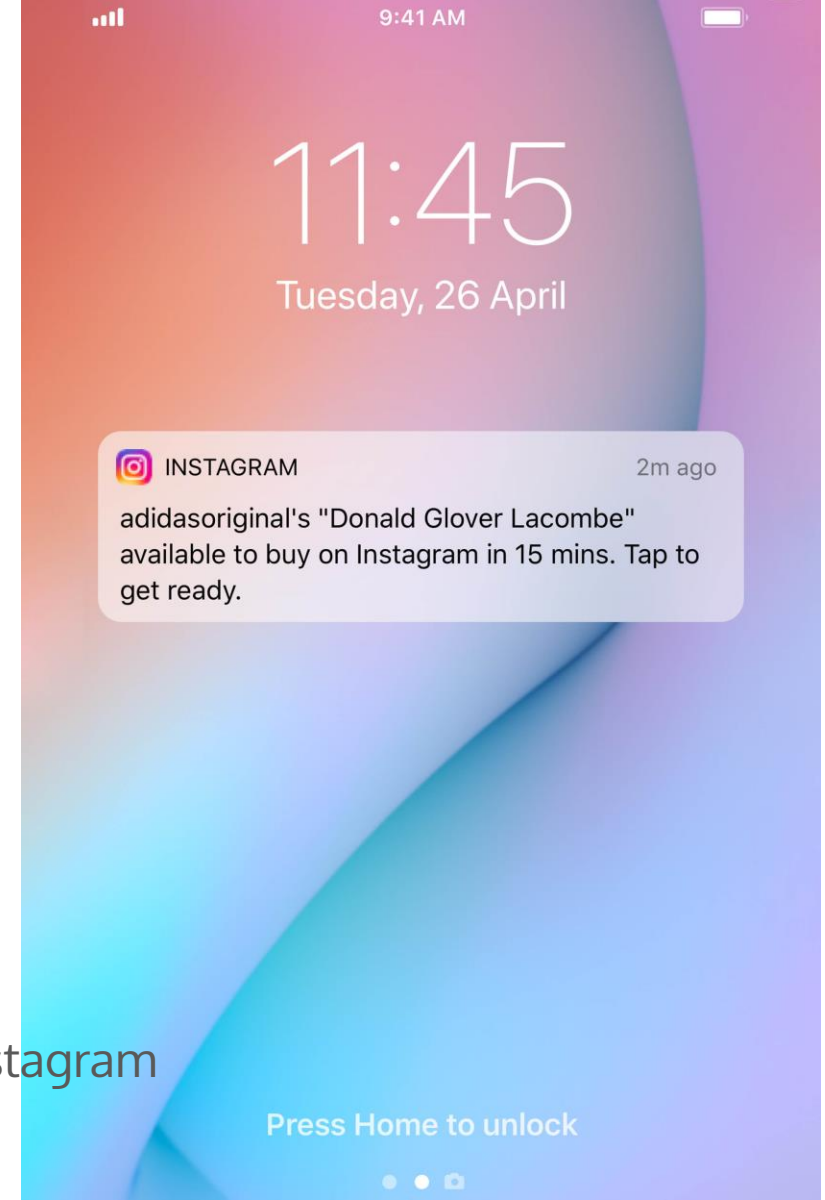
- El **OSNPS** queda esperando "perezosamente" la recepción de una notificación
    - Las reglas de negocio o casos donde se gatilla la notificación son determinadas por la red social.

Recibir notificación

- Cuando se genera el evento, se notifica al **OSNPS**
  - La notificación es enviada al sistema operativo (flujo es ligeramente diferente en Android e iOS).
- Sistema operativo muestra el contenido de la notificación.

Tipos de notificaciones





Notificación recibida en Instagram

Pregunta

¿Cómo puede ir a buscar el timeline histórico en la app de Instagram?

Principios de diseño de eventos

Datos de un evento

Los datos de un evento corresponde a datos de un comportamiento o cambio de estado.

Por lo general, un evento debe contener:

1. Acción (qué está pasando)
2. Timestamp (ya que procesamiento en asíncrono)
3. Estado (otra información relevante del evento
4. Debe ser auto-contenido, toda la información debe estar en el evento.

```
{
    "eventType": "<tipo de evento>",
    "eventTime": "<timestamp>",
    "eventData": {<custom_payload>}
}
```

Datos no son normalizados

Algunos datos se pueden repetir, pero cada evento podría ser diferente.

Puede haber redundancia de datos, contrario a como se usan bases de datos, pero cada evento debe ser autocontenido, no debe depender de otras fuentes de datos*.

*Hay un tradeoff entre el tamaño del payload y la eficiencia de procesamiento del evento.

```
{
    "author": {
        "id": "12345",
        "name": "...",
        "email": "...",
        ...
    }
}
```

```
{
    "authorId": "12345"
}
```

Escalabilidad

El número de eventos producidos puede crecer de un momento a otro exponencialmente.

Los datos guardados en arquitecturas basadas en eventos suelen ser mucho más voluminosos que aquellos en arquitecturas de servicios, ya que los eventos muestran la historia de un sistema mientras que los servicios entregan el estado en un determinado momento.

Ejemplos de eventos reales: AWS Event Bridge


```
{
    "version": "0",
    "id": "0d079340-135a-c8c6-95c2-41fb8f496c53",
    "detail-type": "OrderCreated",
    "source": "myapp.orders",
    "account": "123451235123",
    "time": "2022-02-01T18:41:53Z",
    "region": "us-west-1",
    "detail": {
        "metadata": {
            "correlation_id": "dddd9340-135a-c8c6-95c2-41fb8f492222",
            "service": "my-service",
            "domain": "SHOP"
        },
        "data": {
            "amount": "19.99",
            "quantity": "2",
            "orderId": "0d079340-535a-c8c6-95c2-41fb8f496c53",
            "userId": "9e223efa-e318-4e06-84d3-8734db2f31ea"
        }
    }
}
```

Ejemplos de eventos reales: GCP PubSub


```
{
  "message": {
     "attributes": {
        "Content-Type": "application/json",
        "event_type": "tipo_de_evento"
     },
     "data": "eyJzdGF0dXMiOiAiSGVsbG8gdGhlcmUifQ==",
     "messageId": "2070443601311540",
     "message_id": "2070443601311540",
     "publishTime": "2021-02-26T19:13:55.749Z",
     "publish_time": "2021-02-26T19:13:55.749Z"
  },
  "subscription": "projects/myproject/..."
}
```

Ejemplos de eventos reales: Azure Event Grid


```
{
    "topic": "/subscriptions/{subscription-id}/.../storageAccounts/xstoretestaccount",
    "subject": "/blobServices/default/.../blobs/oc2d2817345i20002296blob",
    "eventType": "Microsoft.Storage.BlobCreated",
    "eventTime": "2017-06-26T18:41:00.9584103Z",
    "id": "831e1650-001e-001b-66ab-eeb76e069631",
    "data": {
      "api": "PutBlockList",
      "clientRequestId": "6d79dbfb-0e37-4fc4-981f-442c9ca65760",
      "requestId": "831e1650-001e-001b-66ab-eeb76e000000",
      ...
      "sequencer": "00000000000004420000000000028963",
      "storageDiagnostics": {
        "batchId": "b68529f3-68cd-4744-baa4-3c0498ec19f0"
      }
    },
    "dataVersion": "",
    "metadataVersion": "1"
}
```

Datos en servicios vs Datos en eventos

	SERVICIOS	EVENTOS
Entidad	Objeto	Hecho o cambio de estado
Esquema	Estricto, cambia poco	Flexible, distintos hechos pueden generar eventos diferentes
Qué es	Una foto del objeto en ese momento del tiempo	Un hecho o cambio de estado en el momento que se produce
Qué describe	Describe el presente	Describe una tendencia en el tiempo
Cómo se actualiza el objeto	Se actualiza, no es necesario mantener historia	Se agrega. Es necesario recorrer toda la historia para ver estado actual.
Complejidad	O(n) Lineal según cantidad de objetos	O(n*k) Según cantidad de objetos (n) y cambios notificados (k)

	Servicios	Eventos
Inicio de la comunicación	Cliente (pull)	Emisor del evento (push)
Participantes	Cliente y servidor	Un emisor, múltiples suscriptores
Sincronía	Comunicación síncrona	Comunicación asíncrona
Uso	Basta con conocer contrato de comunicación para realizar un request	Registro previo del suscriptor, quién notifica eventos en la medida que suceden
Tipo de datos	Estado actual de objetos	Estado en un cierto momento
Arquitectura	Cliente - servidor	Emisor - Event broker - Suscriptores

Arquitecturas basadas en eventos se ven interesantes, pero los servicios parecieran estar en todas partes

Especialmente en el mundo web

SERVICIOS

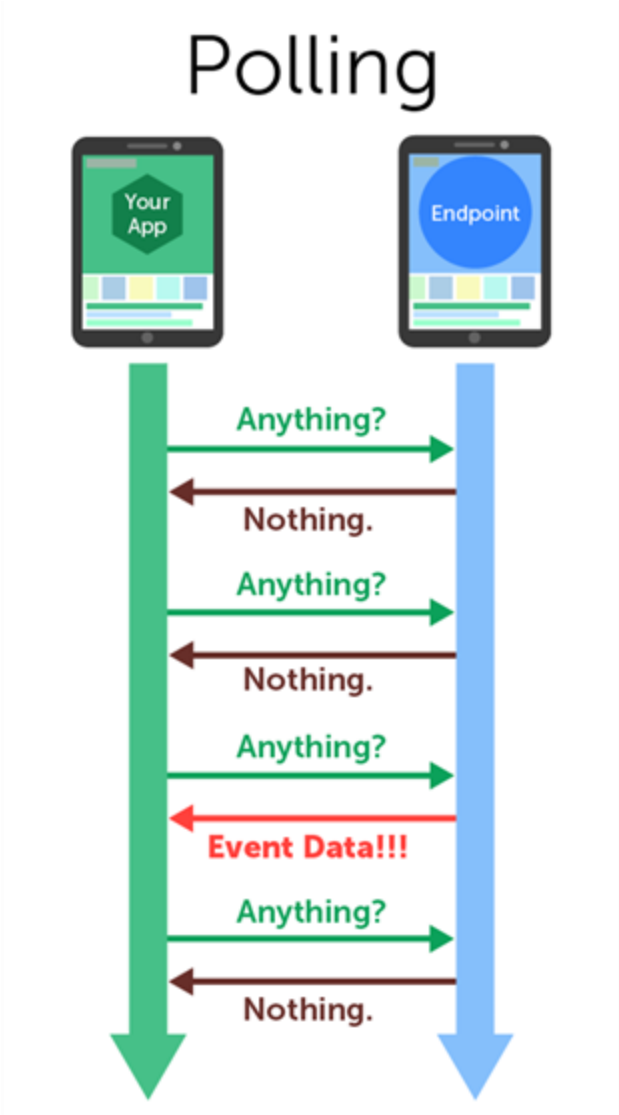
EVENTOS

Eventos en el mundo web

Obteniendo cambios de estados desde un servicio

Técnica de Polling (o “long polling”):

Consultar constantemente por el estado hasta que haya algún cambio.



https://es.mailjet.com/blog/news/que-es-webhook/

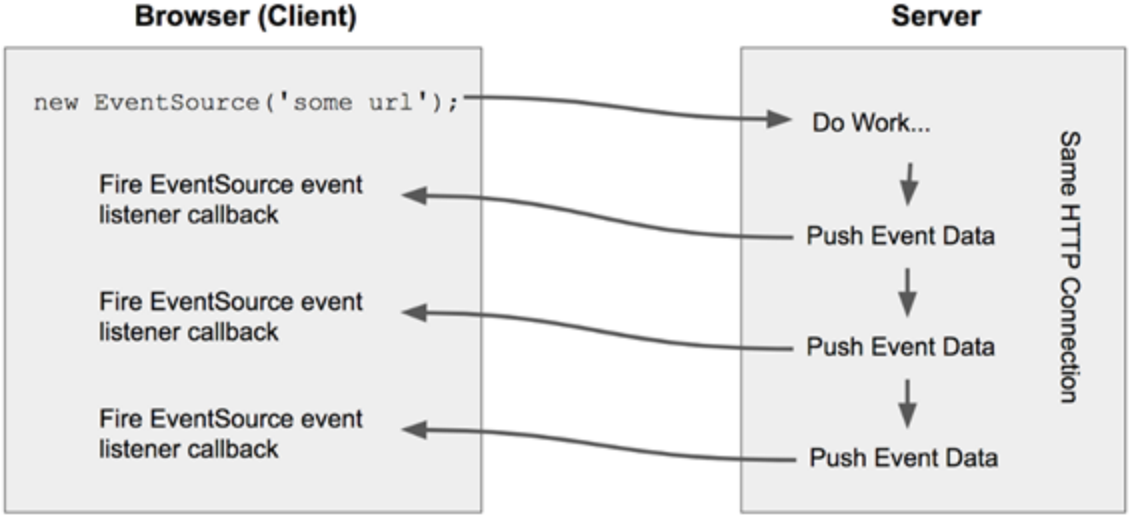
Server-sent events

Estándar web, incluido en HTML5, para notificar eventos al navegador.

Soportado por casi todos los navegadores (menos IE 🤦) y servidores web.

Operan sobre HTTP. Se crea una conexión y se mantiene abierta para recibir nuevos datos.

Se puede usar de servidor a navegador o servidor a servidor.



Websockets

Protocolo de comunicación bidireccional, por sockets que operan a través de internet.

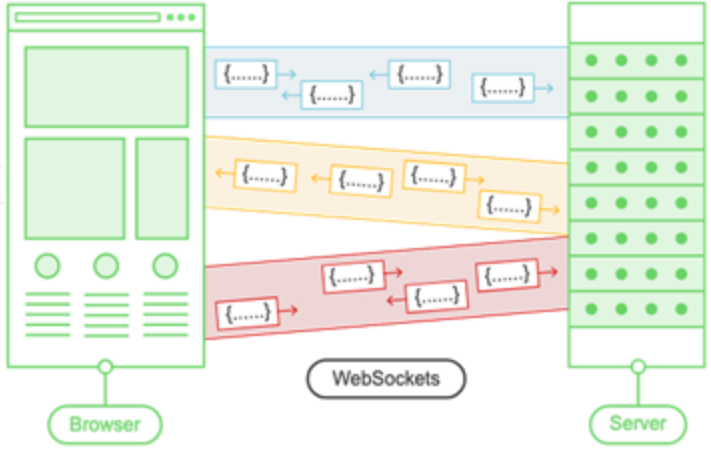
La comunicación es bidireccional, lo que hace que el protocolo sea muy apropiado para contextos de comunicación en tiempo real.

Se puede usar de servidor a navegador o servidor a servidor.

Requiere de frameworks o librerías adicionales a los servidores tradicionales. Algunos servidores web no soportan websockets (AWS Lambda incorporó soporte hace poco, GCP Cloud Functions no tiene soporte).

Soportado por todos los navegadores web.

La comunicación es más liviana que en operaciones HTTP.



Webhooks

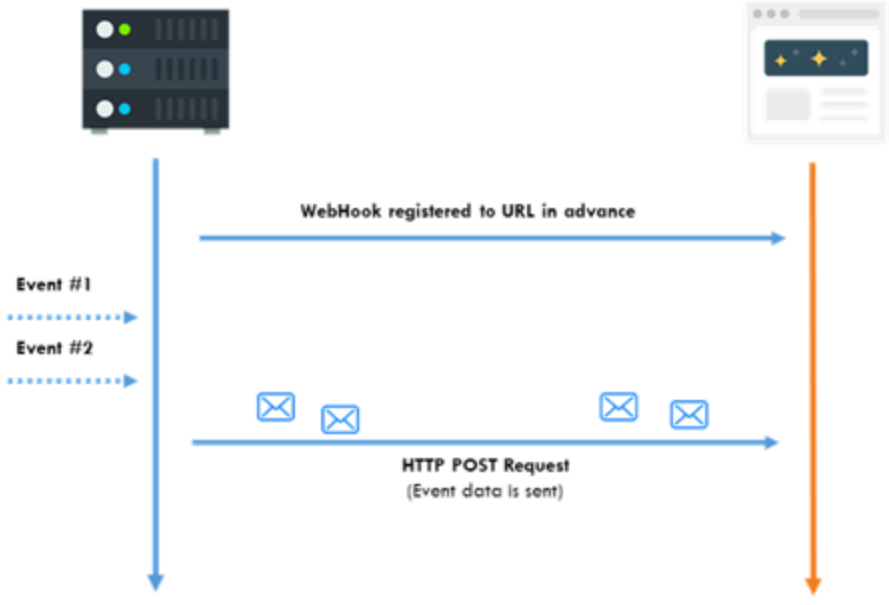
Notificación de un evento a través de un servicio, utilizado especialmente en servidores web. Método de comunicación servidor a servidor.

En vez de que nuestro servidor esté realizando un polling a un servicio, quién gatilla el evento nos notifica del cambio esperado.

Se debe crear un servicio de acuerdo a la definición del emisor del evento.

Quién inicia la comunicación es el servidor (emisor de evento). (Sin considerar la configuración inicial)

Se debe suscribir el servicio con quién emite el evento. Puede haber múltiples destinatarios para un mismo evento.



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IIC3103

Taller de Integración

Profesores

Arturo Tagle / Daniel Darritchon

Bibliografía

Eventos

[Architecting Event-driven Systems the right way | by Jayanth Kumar]

https://blog.axway.com/learning-center/apis/basics/event-driven-vs-rest-api-interactions

https://www.boyney.io/blog/2022-02-11-event-payload-patterns

https://cloud.google.com/pubsub/docs/payload-unwrapping?hl=es-419

https://learn.microsoft.com/en-us/azure/event-grid/event-schema

Bibliografía

Principios de diseño de eventos

Wetzler, M. (2020, July 1). Analytics for hackers: How to think about event data - keen - event streaming platform. Keen.io blog. Retrieved September 27, 2021, from https://keen.io/blog/analytics-for-hackers-how-to-think-about-event-data/ .

Higginbotham, J. (2018, May 2). 5 principles for designing evolvable event streams. Medium. Retrieved September 27, 2021, from https://medium.com/capital-one-tech/5-principles-for-designing-evolvable-event-streams-f32e90dcbb79 .