



Dynamics AX – Integración de aplicaciones

www.maint.com.ec





Integración a Dynamics

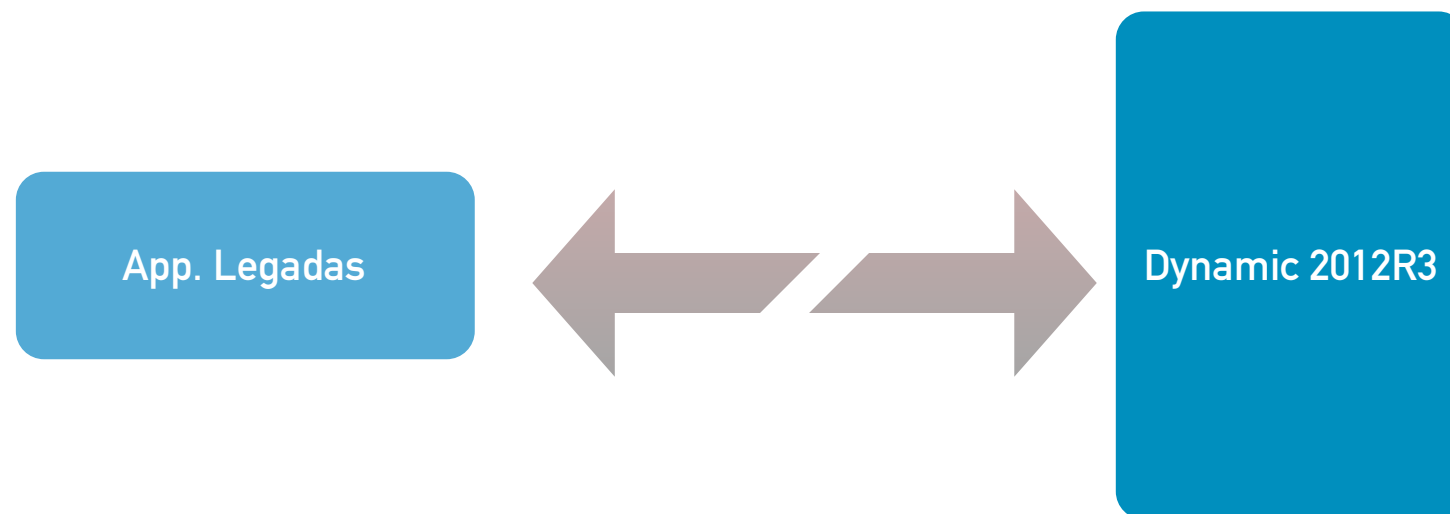


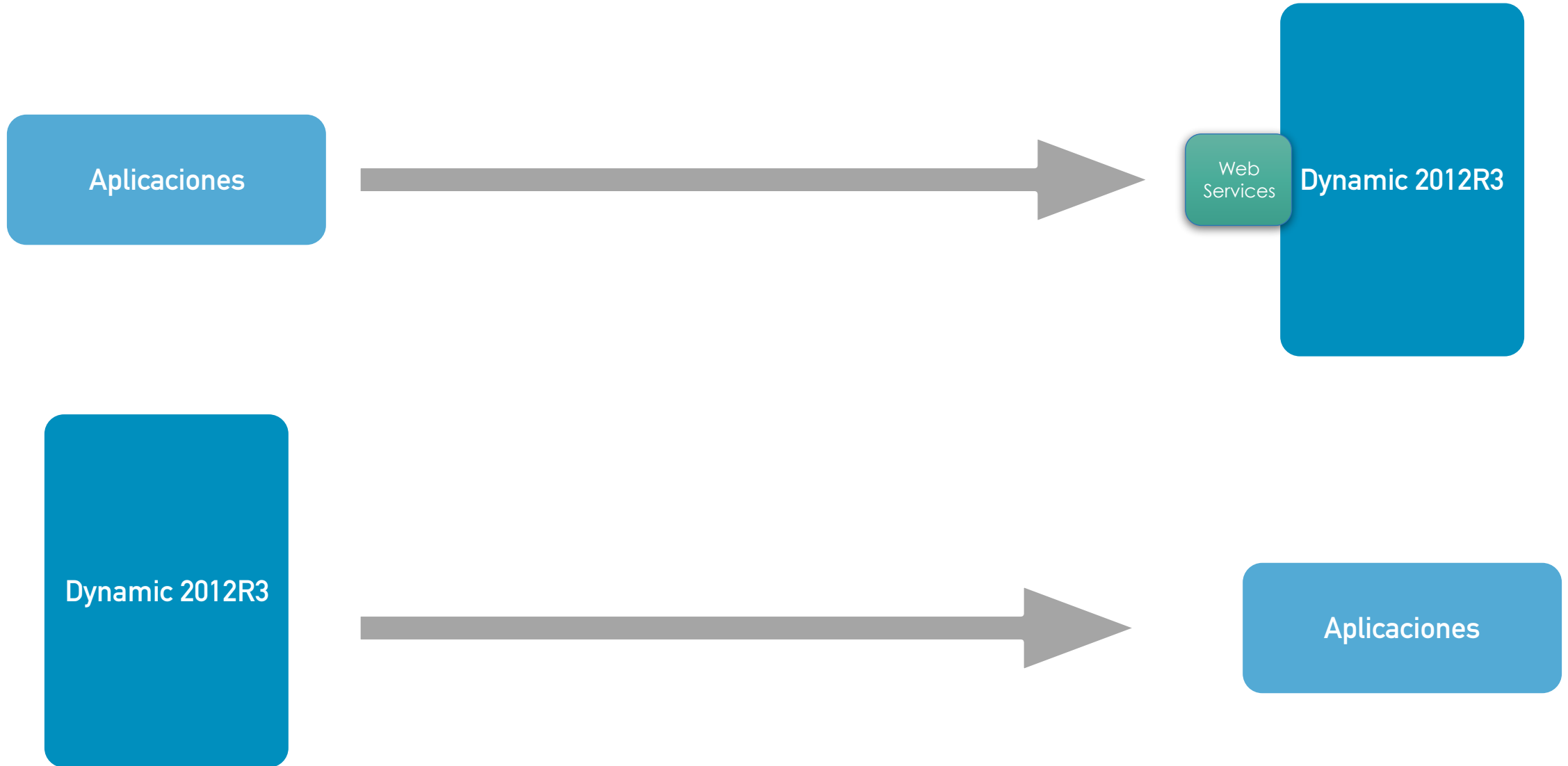


situación actual

- Sistemas legados que necesitan integrarse
- Liberación de funcionalidad de AX mediante Web services
- Administración de carga de procesamiento









Problemática a resolver

Priorización

Balanceo de carga

Interfaces síncronas / asíncronas

Planeación de tareas en lote

Estandarización de interface





¿Qué es una solución de colas de mensajes?

Dentro de un proyecto en ocasiones hay que integrarse con otros actores, componentes o sistemas internos y externos, surgiendo la necesidad de aportar o recibir información de ellos.

En el mayor de los casos, estas comunicaciones tienen que:

- estar permanente **disponibles**,
- ser **rápidas**,
- **seguras**
- **asíncronas** y
- **fiables**



Las colas de mensajes solucionan estas necesidades actuando de middleware entre emisores y destinatarios, o en un contexto más definido, productores y consumidores de mensajes. Aportan a su vez más beneficios:

- Garantía de entrega y orden
- Redundancia
- Desacoplamiento
- Escalabilidad



Beneficios



Garantía de entrega y orden

Los mensajes se consumen, en el mismo orden que llegaron a la cola, y son consumidos una única vez.



Desacoplamiento

Siendo capas intermedias de comunicación entre procesos, aportan la flexibilidad en la definición de arquitectura de cada uno de ellos de manera separada, siempre que se mantenga una interfaz común.



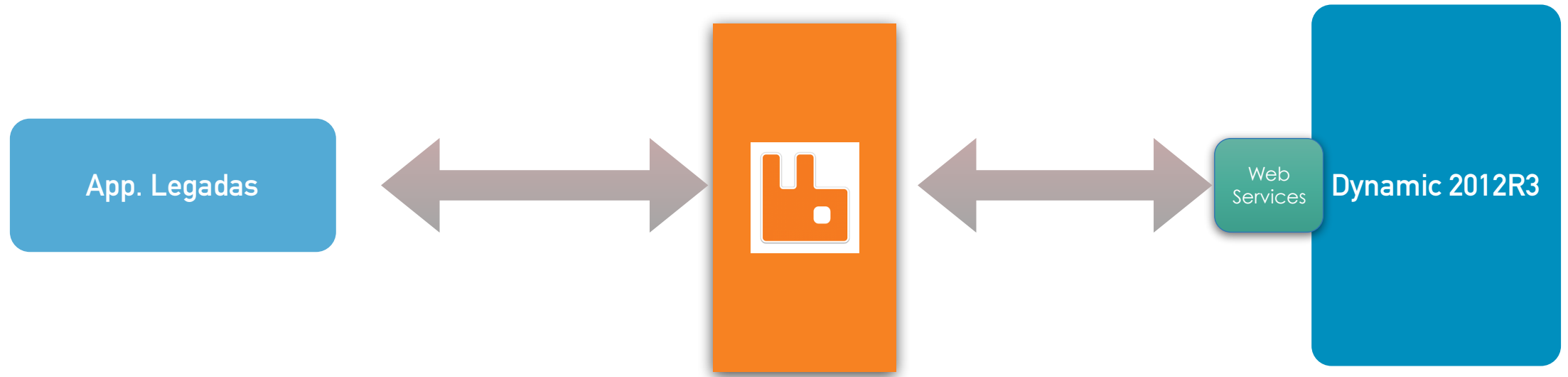
Redundancia

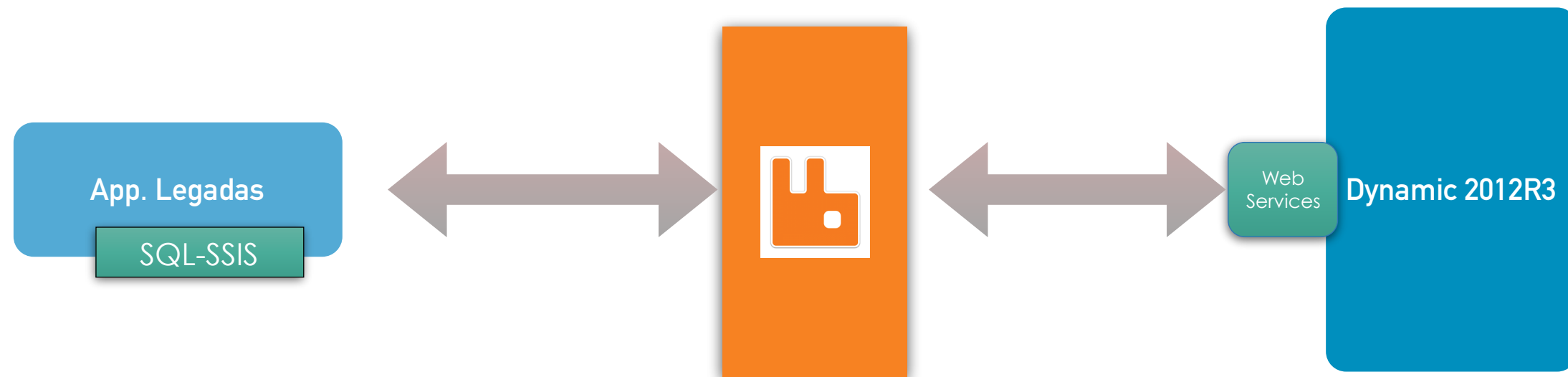
Las colas persisten los mensajes hasta q son procesados por completo.

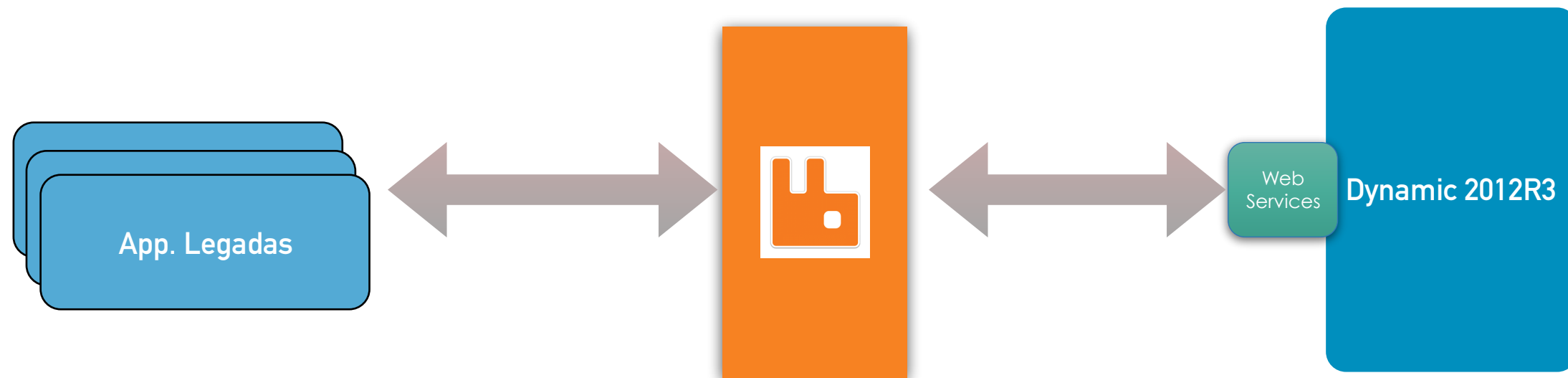


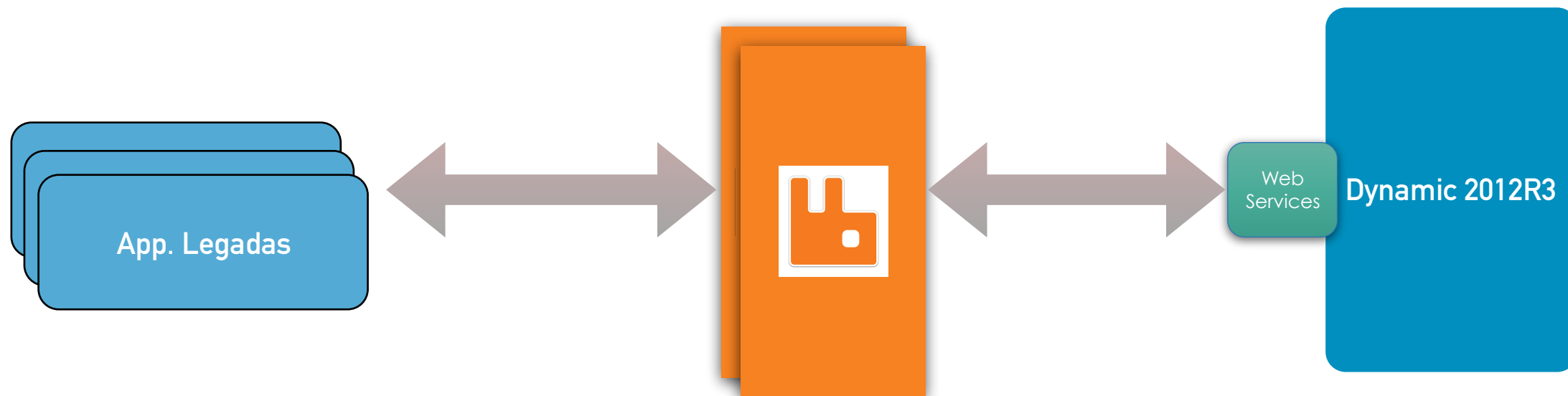
Escalabilidad

con más unidades de procesamiento, las colas balancean su respectiva carga











¿A DÓNDE QUEREMOS LLEGAR?





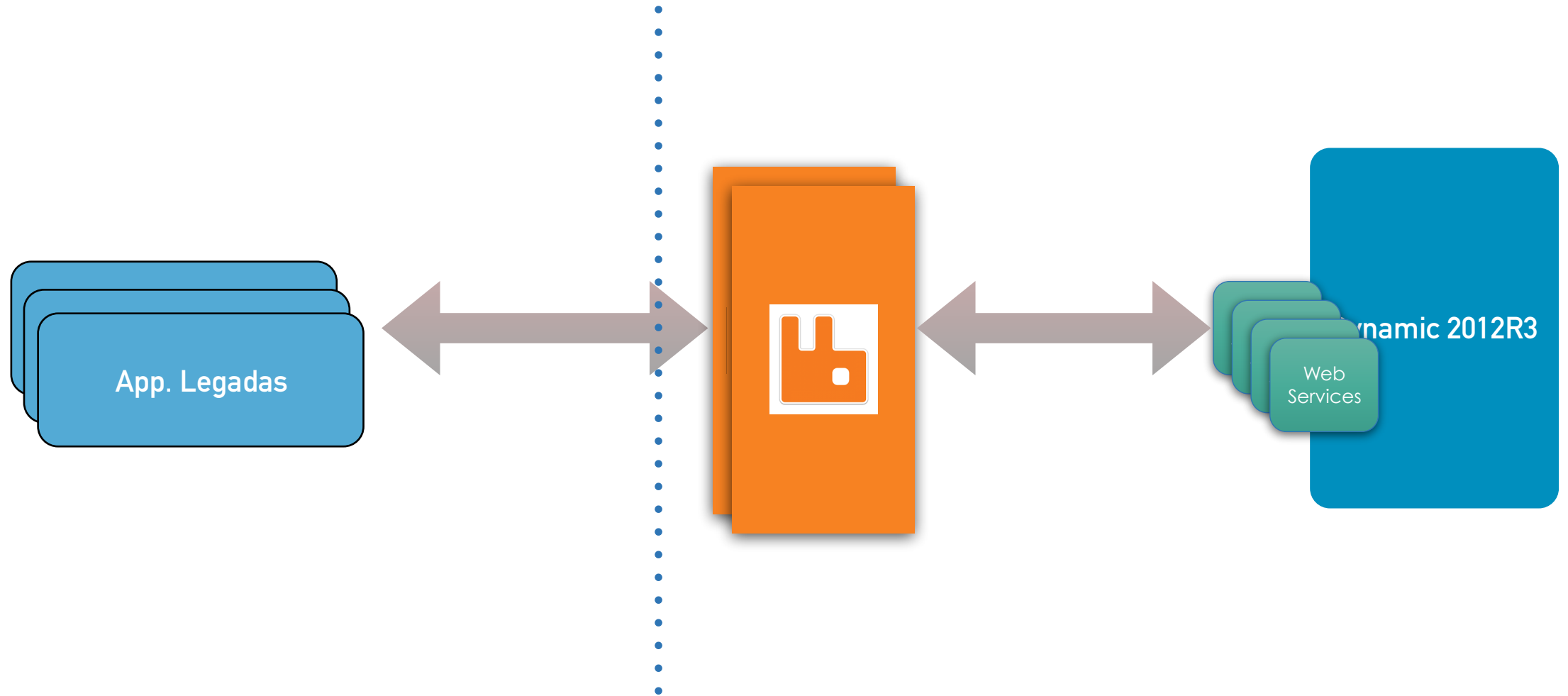
RabbitMQ

RabbitMQ, un sistema de mensajería empresarial completo y altamente confiable basado en el estándar AMQP.

Sus características principales son :

- Garantía de entrega
- Enrutamiento flexible
- Alta disponibilidad
- Tolerancia a fallos







Funcionalidad a implementar

- Comunicación síncrona , asíncrona(opcional)
- Balanceo de carga
- Estandarización de interfaces con sistemas legados
- Planificación de tareas en lote
- Visibilidad de la operación de las interfaces
- Simplicidad en la integración





Simplicidad en la integración

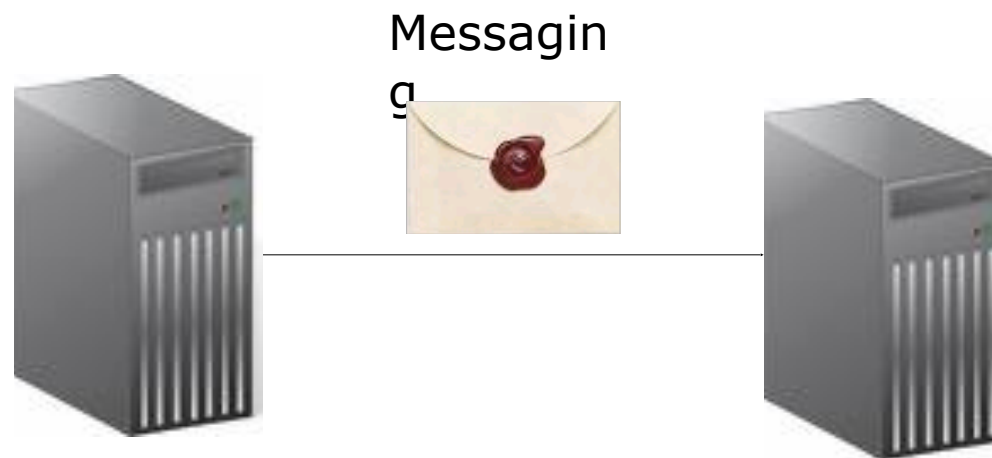
- Pocas líneas de código para integrar – receta estándar



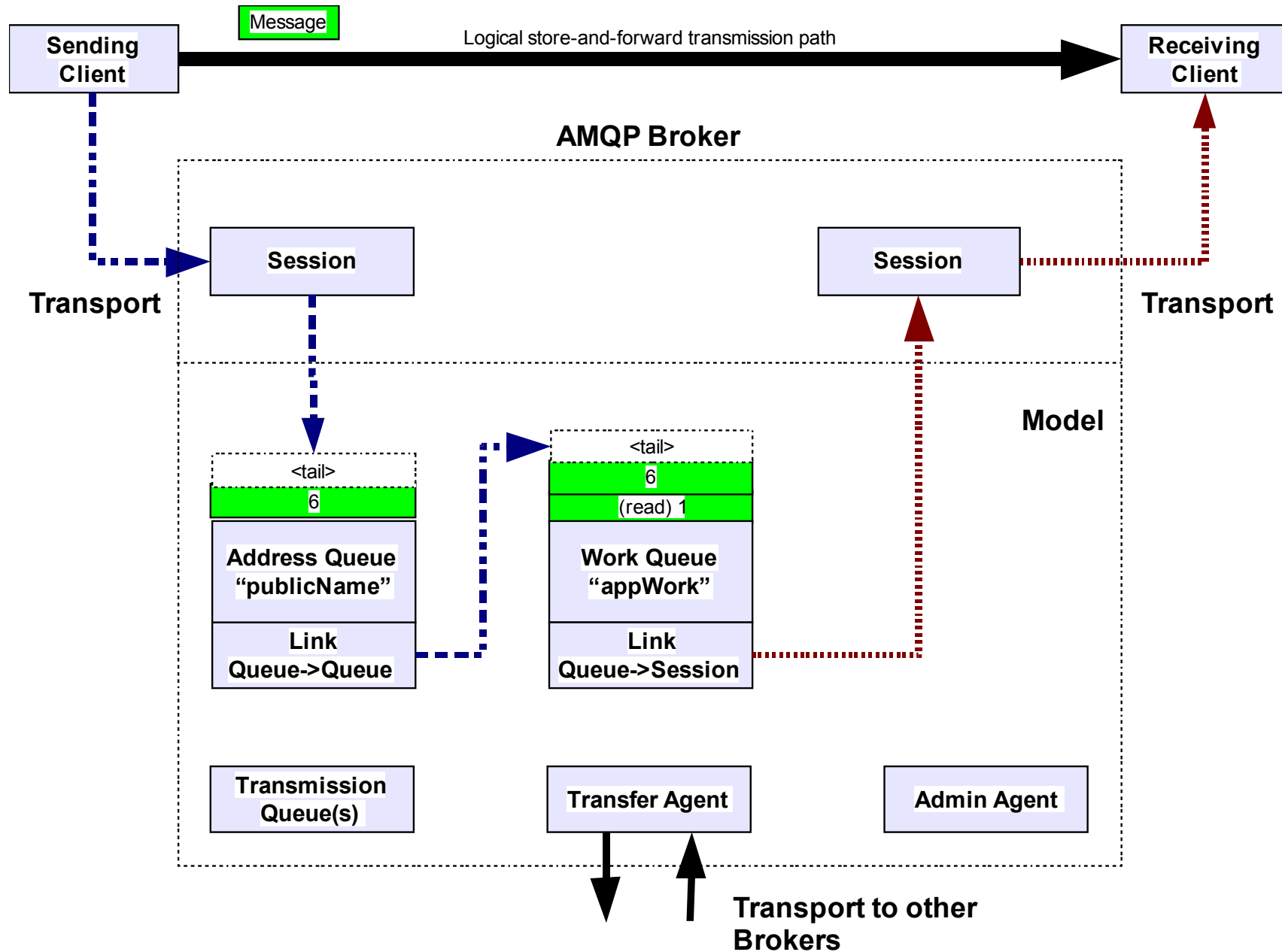
¿Qué es AMQP?



AMQP

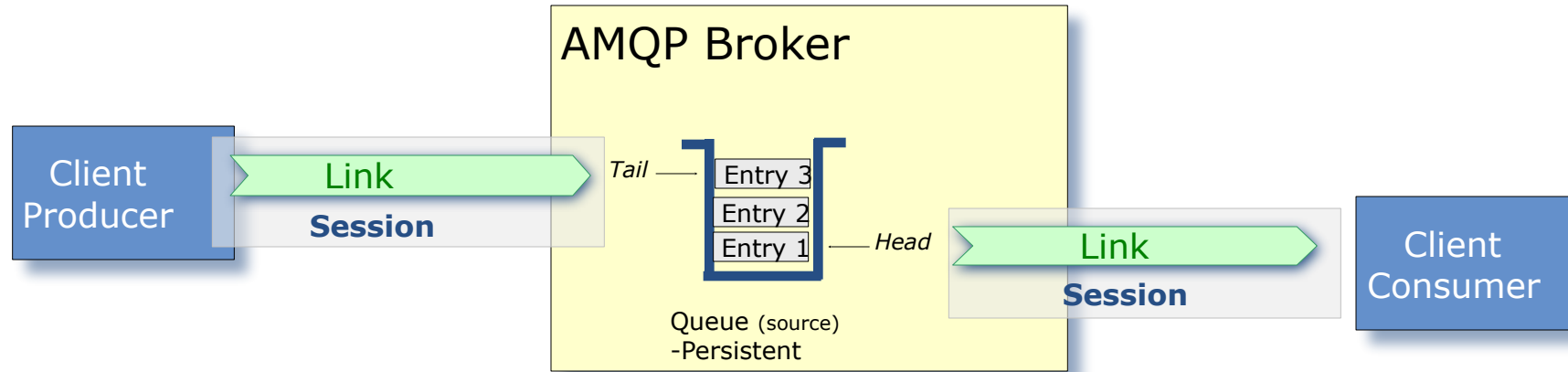


AMQP Data Flow Overview





Point-to-point Queue



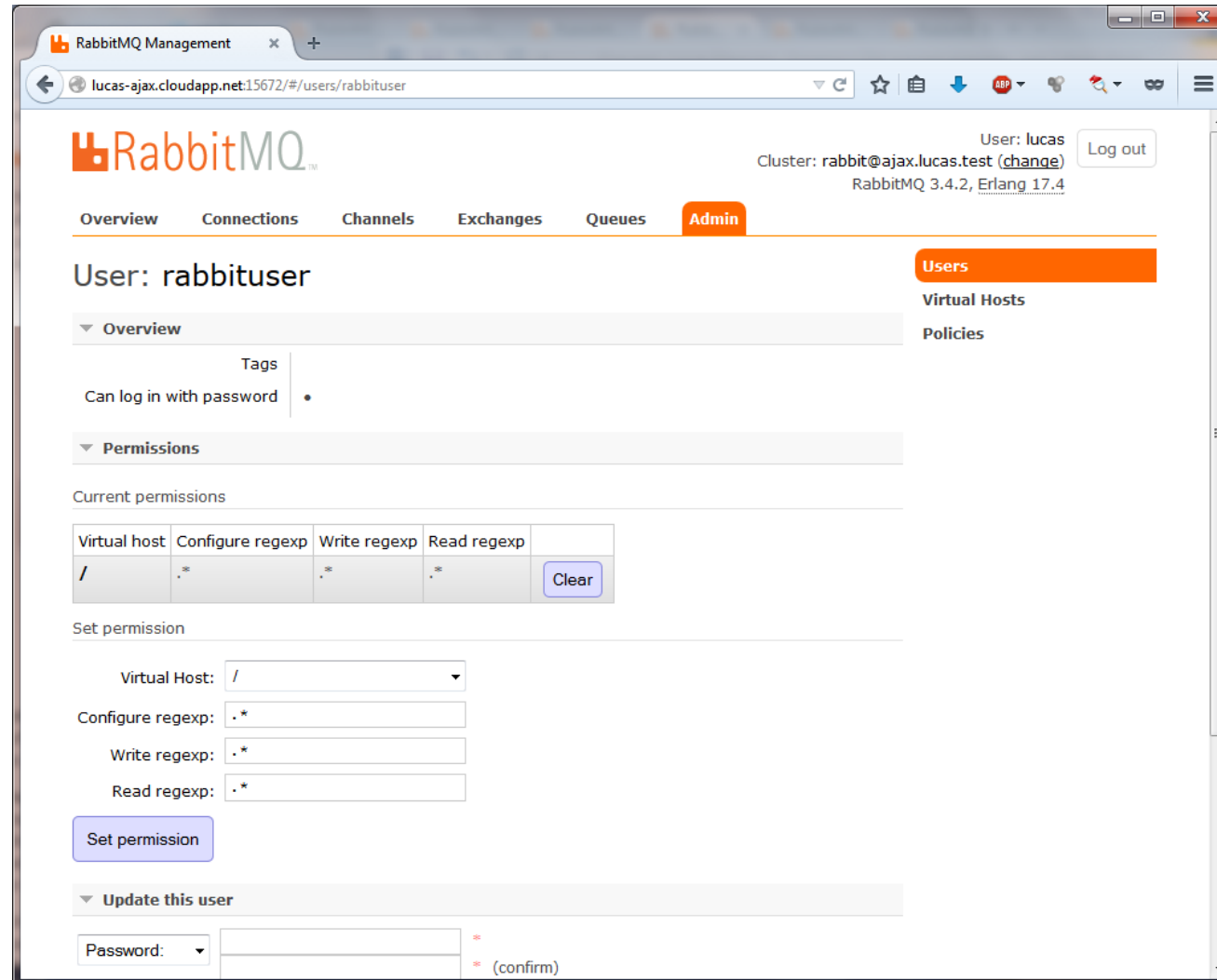


¿Administración RabbitMQ?



RabbitMQ Administración - Usuarios

http://localhost:15672



The screenshot shows the RabbitMQ Management web interface in a browser. The address bar indicates the URL is `lucas-ajax.cloudapp.net:15672/#/users/rabbituser`. The page title is "RabbitMQ Management". In the top right corner, it shows "User: lucas" with a "Log out" button, and "Cluster: rabbit@ajax.lucas.test (change)" and "RabbitMQ 3.4.2, Erlang 17.4".

The main navigation bar includes "Overview", "Connections", "Channels", "Exchanges", "Queues", and "Admin" (which is highlighted). Below this, the page title is "User: rabbituser".

On the right side, there is a sidebar with links: "Users" (highlighted in orange), "Virtual Hosts", and "Policies".

The main content area has a "Tags" section with "Can log in with password" and a "Permissions" section. Under "Permissions", there is a table showing "Current permissions" for the virtual host `/`. The table has columns for "Virtual host", "Configure regexp", "Write regexp", and "Read regexp", all containing the value `.*`. There is a "Clear" button next to the table.

Below the table, there is a "Set permission" section with input fields for "Virtual Host" (set to `/`), "Configure regexp" (set to `.*`), "Write regexp" (set to `.*`), and "Read regexp" (set to `.*`). There is a "Set permission" button.

At the bottom, there is an "Update this user" section with a "Password:" label and two input fields, one of which is marked with a red asterisk and labeled "(confirm)".

RabbitMQ Administración - Colas

RabbitMQ Management

lucas-ajax.cloudapp.net:15672/#/queues

User: lucas
Cluster: rabbit@ajax.lucas.test (change)
RabbitMQ 3.4.2, Erlang 17.4

Log out

Overview Connections Channels Exchanges **Queues** Admin

Queues

▼ All queues

Filter: ☐ Regex (?) 3 items (show at most)

Overview			Messages			Message rates		
Name	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
CRM-Cases	D	idle	0	0	0	0.00/s	0.00/s	
CRM-Leads	D	idle	0	0	0	0.00/s	0.00/s	0.00/s
CRM_Demo	D	idle	0	0	0	0.00/s	0.00/s	

▼ Add a new queue

Name:

Durability:

Auto delete: (?)

Arguments: =

Add Message TTL (?) | Auto expire (?) | Max length (?) | Max length bytes (?)
Dead letter exchange (?) | Dead letter routing key (?)

Add queue

HTTP API | Command Line

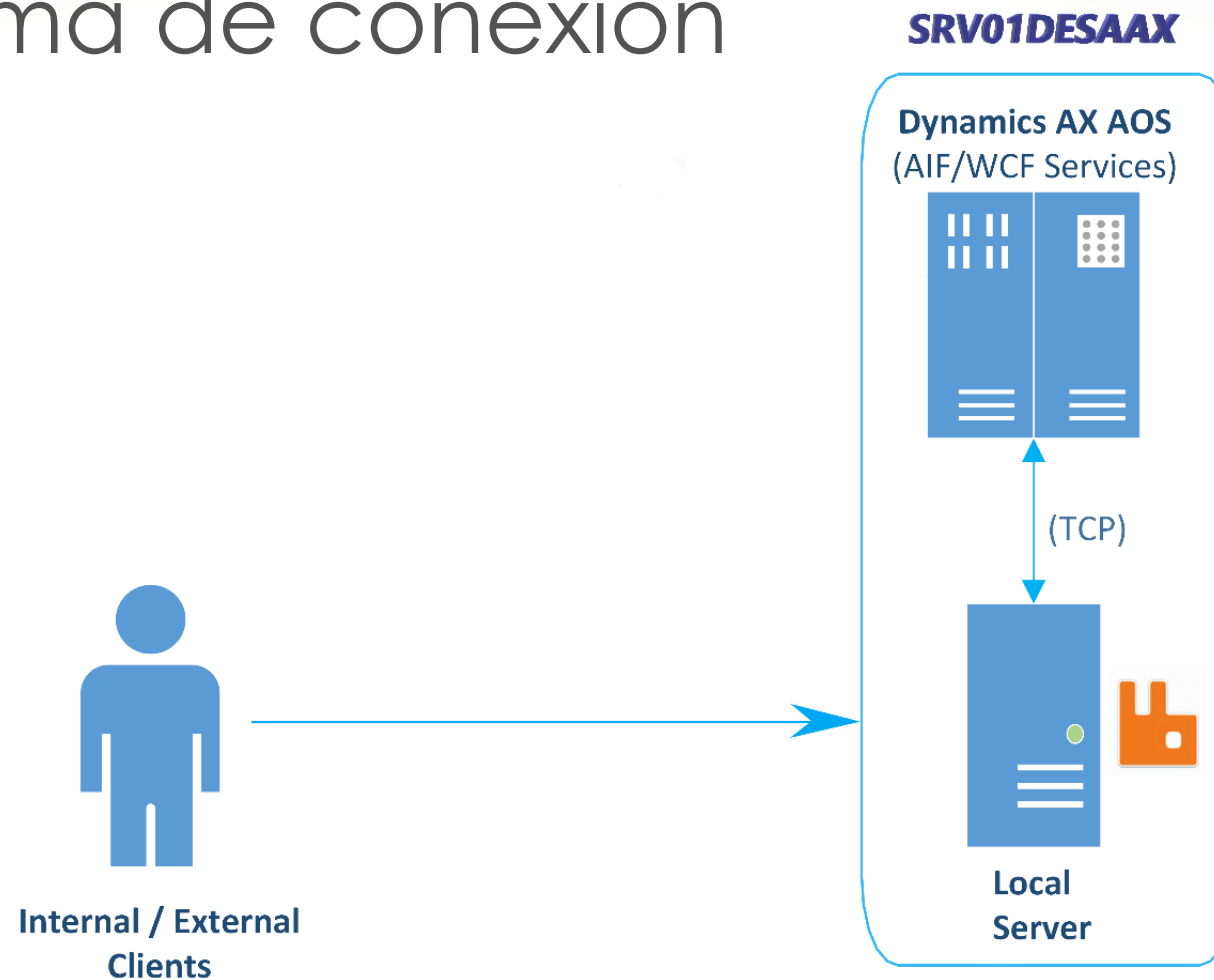
Update

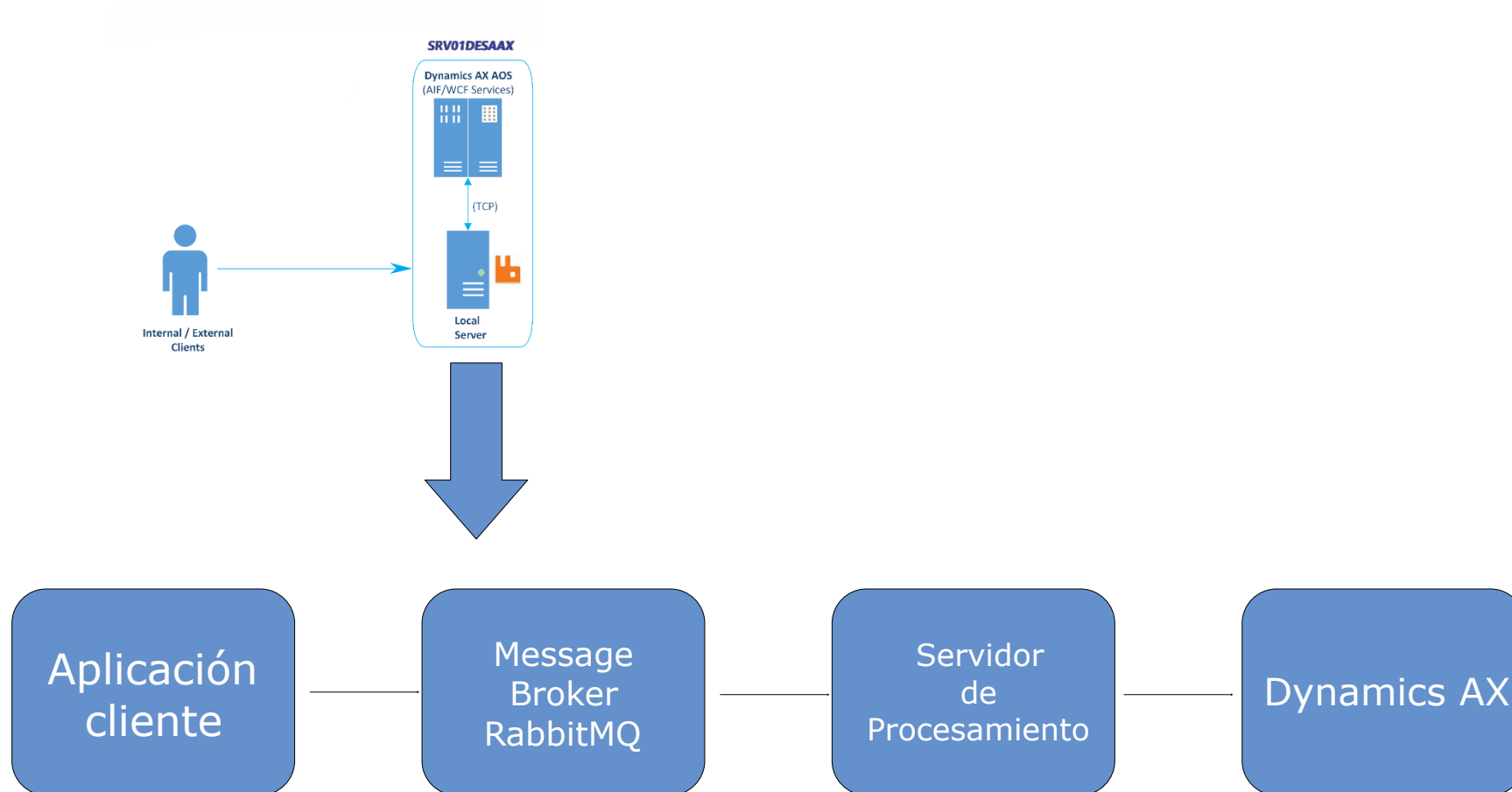
Last updated: 2014-12-20 11:10:52

Escenario CRECOS

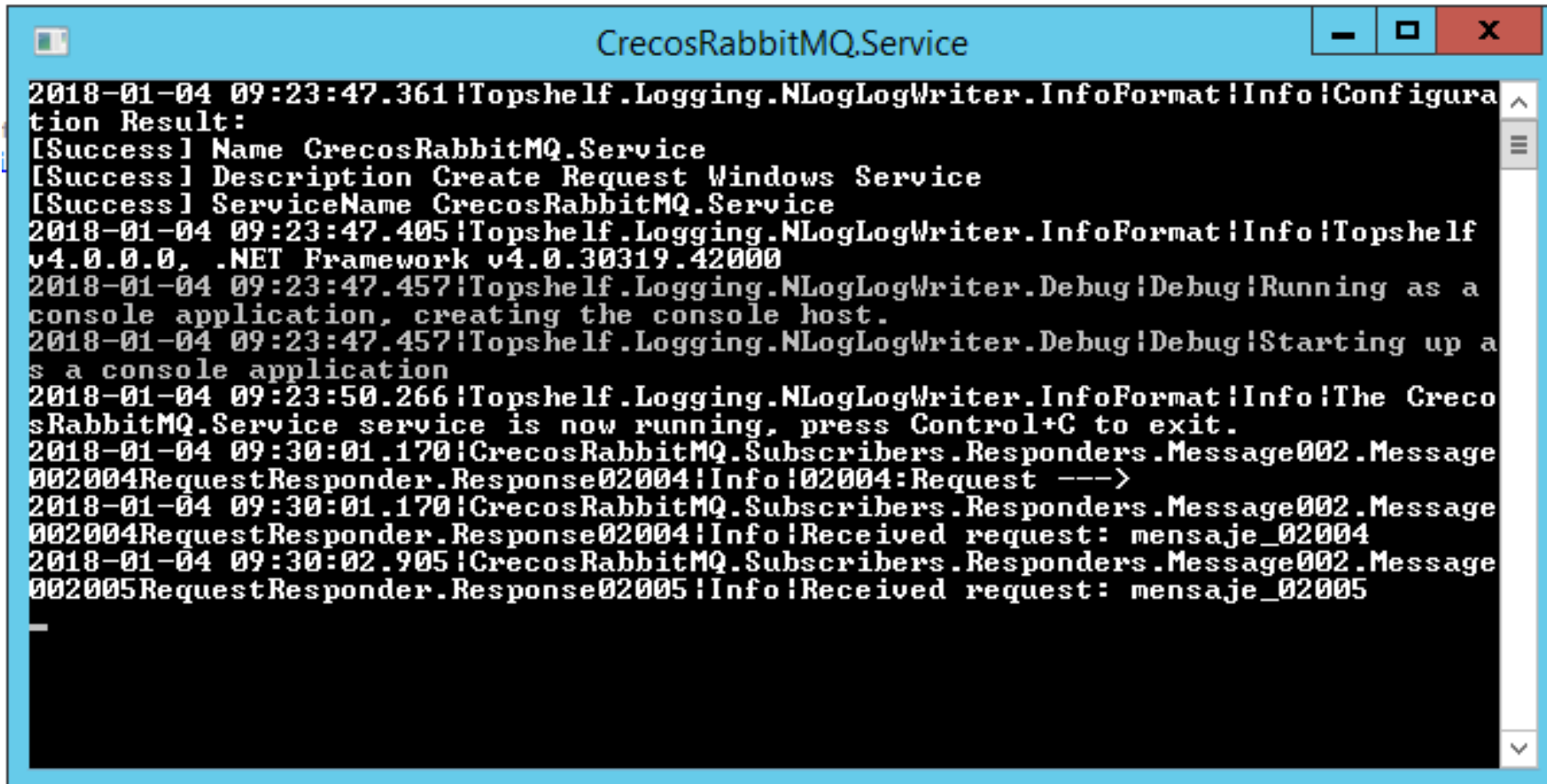


Esquema de conexión





Servidor de Procesamiento



```
2018-01-04 09:23:47.361!Topshelf.Logging.NLogLogWriter.InfoFormat!Info!Configura
tion Result:
[Success] Name CrecosRabbitMQ.Service
[Success] Description Create Request Windows Service
[Success] ServiceName CrecosRabbitMQ.Service
2018-01-04 09:23:47.405!Topshelf.Logging.NLogLogWriter.InfoFormat!Info!Topshelf
v4.0.0.0, .NET Framework v4.0.30319.42000
2018-01-04 09:23:47.457!Topshelf.Logging.NLogLogWriter.Debug!Debug!Running as a
console application, creating the console host.
2018-01-04 09:23:47.457!Topshelf.Logging.NLogLogWriter.Debug!Debug!Starting up a
s a console application
2018-01-04 09:23:50.266!Topshelf.Logging.NLogLogWriter.InfoFormat!Info!The Crecos
RabbitMQ.Service service is now running, press Control+C to exit.
2018-01-04 09:30:01.170!CrecosRabbitMQ.Subscribers.Responders.Message002.Message
002004RequestResponder.Response02004!Info!02004:Request --->
2018-01-04 09:30:01.170!CrecosRabbitMQ.Subscribers.Responders.Message002.Message
002004RequestResponder.Response02004!Info!Received request: mensaje_02004
2018-01-04 09:30:02.905!CrecosRabbitMQ.Subscribers.Responders.Message002.Message
002005RequestResponder.Response02005!Info!Received request: mensaje_02005
```

Dynamics AX – Web Services: Puertos de Entrada

Puertos de entrada (1)

Archivo Nuevo Eliminar Activar Desactivar

Nombre del puerto	Descripción
✓ mntEmpleado	
✓ mntFactSaldoService...	
✗ mntFacturasServicios	Generación automát...
✓ mntIndiceComerGen	
✓ mntIndiceComerSku	
✓ mntIndiceFinanSku	
✓ mntOrdenTrabajo	
✓ mntOrdenTransfere...	mntOrdenTransfere...
✓ mntOrdenVenta	
✓ OMLegalEntityServic...	
✗ PayrollServices	
✗ PriceDiscServices	
✗ PurchRFQ	
✗ ReqDemPlanServices	
✓ RetailRapidStart	
✓ RetailServices	

Nombre del puerto: AccountsPayableServices

Descripción:

Categoría: Base *Grupo de servicio definido por el desarrollador*

Dirección

Adaptador: NetTcp

URI: net.tcp://AOS_SERVICE_HOST/DynamicsAx/Services/AccountsPayableServices

Solución de problemas Registro deshabilitado

Modo de registro: Registro deshabilitado

Nombre del puerto



Código: Aplicación cliente

Código – Grupo de Transacciones:

001 Cliente

002 Orden de Trabajo

003 Orden de Transferencia

004 Casos

005 RRHH

006 Orden de Venta

007 Consulta Saldos Facturas



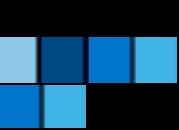
Tipos de Mensajes

No.	Descripción
001	Crear Orden de Trabajo
002	Actualizar Orden de Trabajo
003	Consultar Orden de Trabajo por código del cliente
004	Consultar Item Servicios
005	Consultar requisitos y materiales de instalación del artículo.
006	Consulta cliente por código

```
try
{
    var myOrdenTrabajo = JsonConvert.DeserializeObject<OrdenTrabajo>(richTextBox3.Text);
    var response = new MessageResponse002001();
    OrdenTrabajoValidator validator = new OrdenTrabajoValidator();
    ValidationResult results = validator.Validate(myOrdenTrabajo);

    bool validationSucceeded = results.IsValid;
    IList<ValidationFailure> failures = results.Errors;
    var myList = new ErrorList[failures.Count];
    var i = 0;
    foreach (ValidationFailure data in failures)
    {
        myList.SetValue(new ErrorList { Code = "" + failures.IndexOf(data), Descripcion = data.ErrorMessage }, i);
        i++;
    }
    if (validationSucceeded)
    {
        var myRequest = new MessageRequest002001 {
            header = Header.mensaje_02001,
            prioridad = Prioridad.Alta,
            ordenTrabajo = myOrdenTrabajo,
            credencial = SetSecurity(),
            empresa = "CREC"
        };
        response = bus.Request<MessageRequest002001, MessageResponse002001>(myRequest);
        richTextBox1.Text = response.response;
        textBoxError.Text = response.errorCode;
    }
    else
    {
        textBoxError.Text = "Error de validación";
    }
}
catch (Exception ex)
{
    richTextBox1.Text = ex.InnerException.Message;
    textBoxError.Text = "";
}
```


RabbitMQ + LDAP



RABBITMQ + LDAP

- enable RABBITMQ LDAP PLUGIN

\$~ rabbitmq-plugins enable rabbitmq_auth_backend_ldap

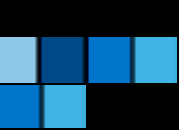


RABBITMQ + LDAP

- config RABBITMQ auth backend

```
{rabbit,[{auth_backends,[rabbit_auth_backend_ldap]}]}
```

```
{rabbit,[{auth_backends,  
          [rabbit_auth_backend_ldap,rabbit_auth_backend_internal]}]}
```



RABBITMQ + LDAP

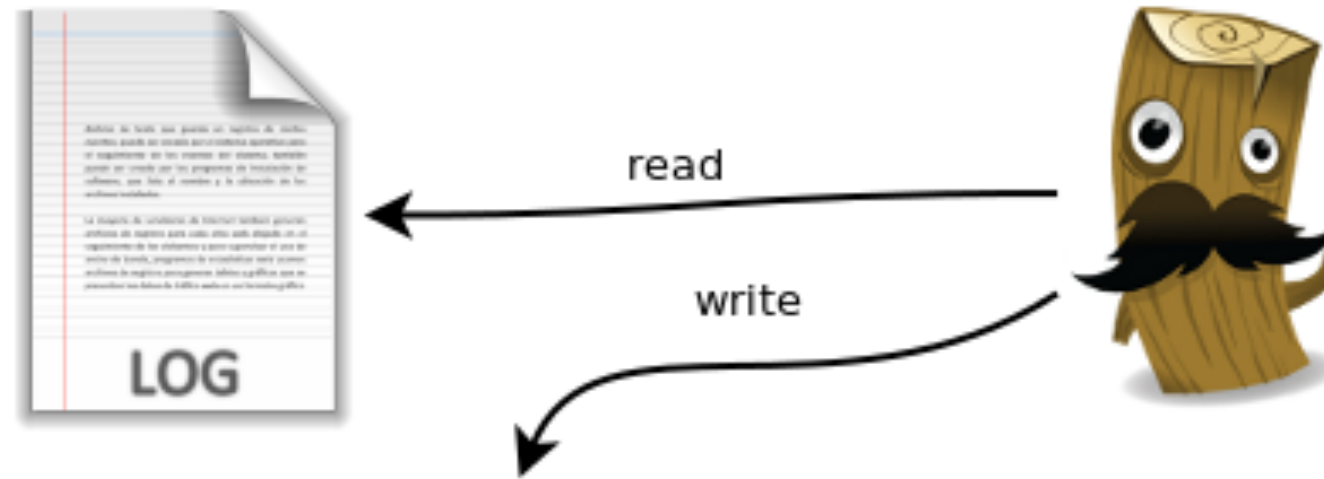
- config RABBITMQ LDAP plugin (example)

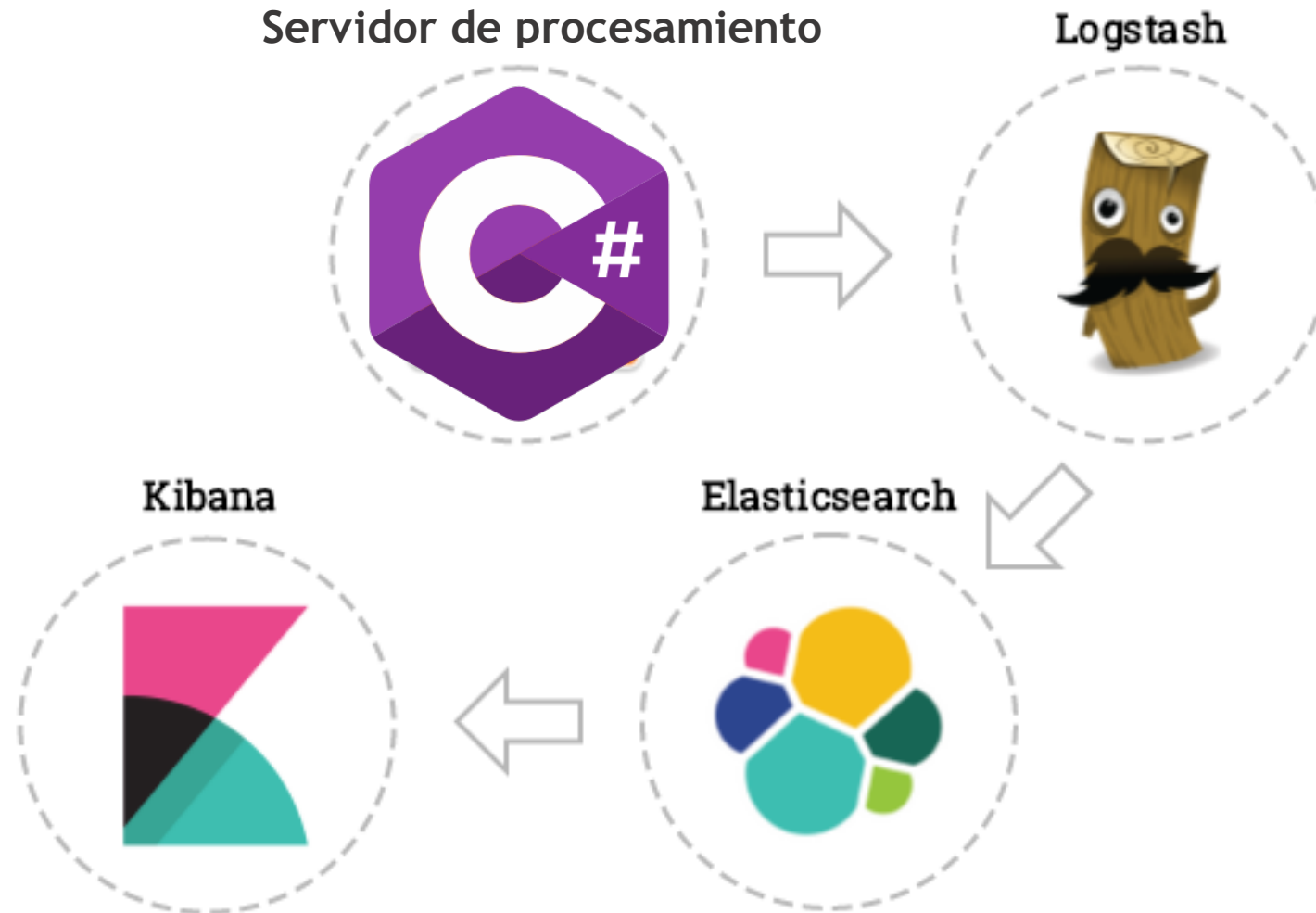
[...]

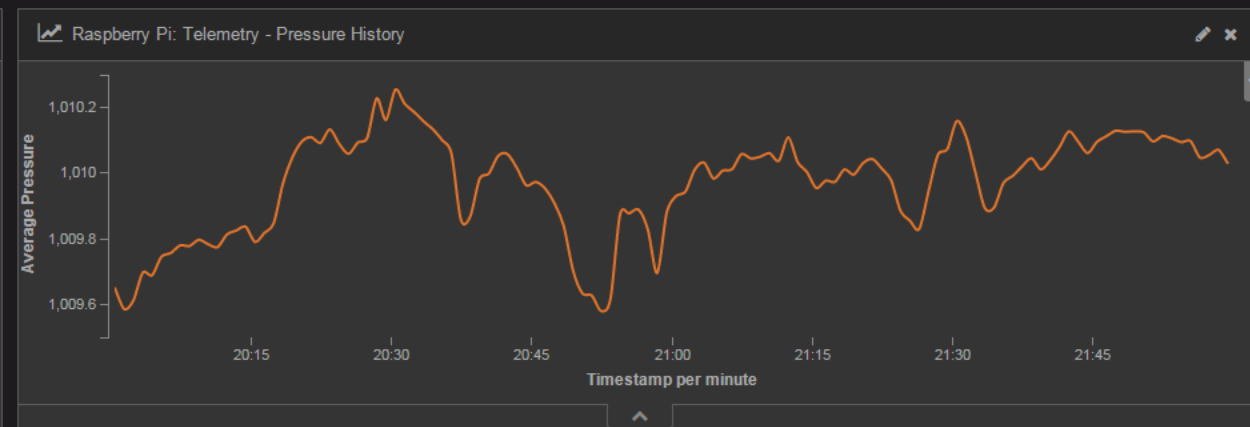
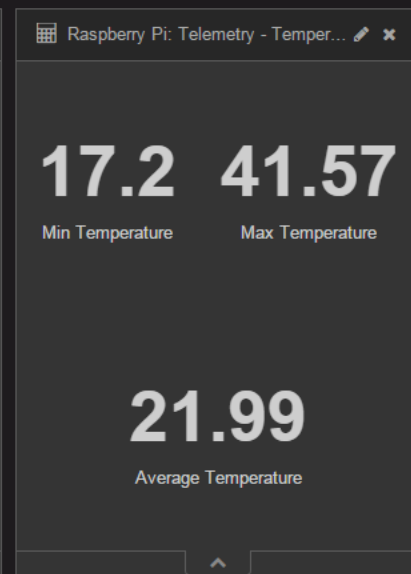
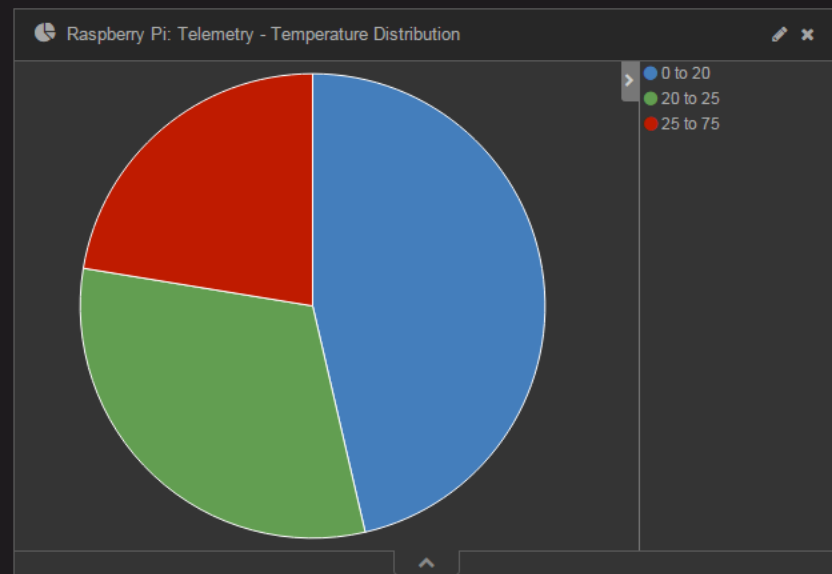
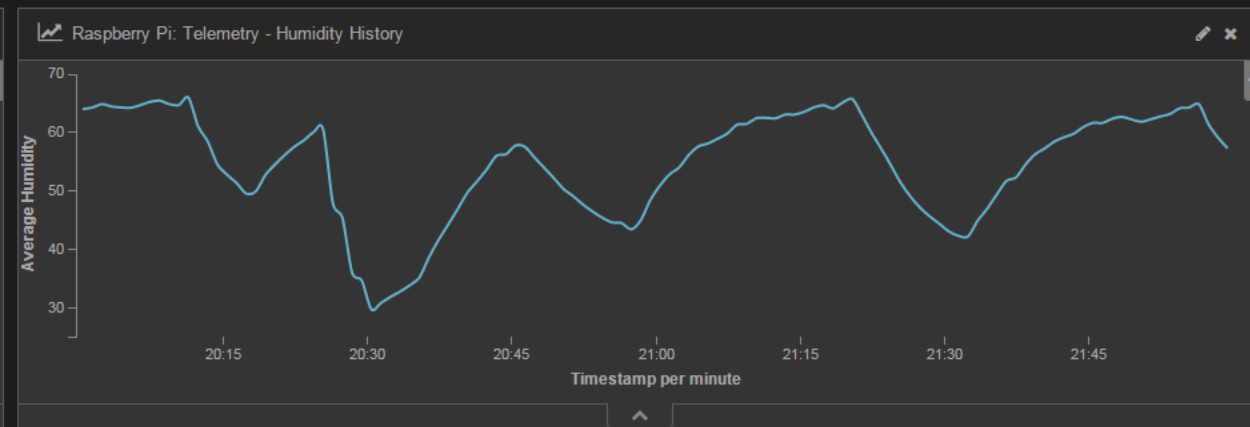
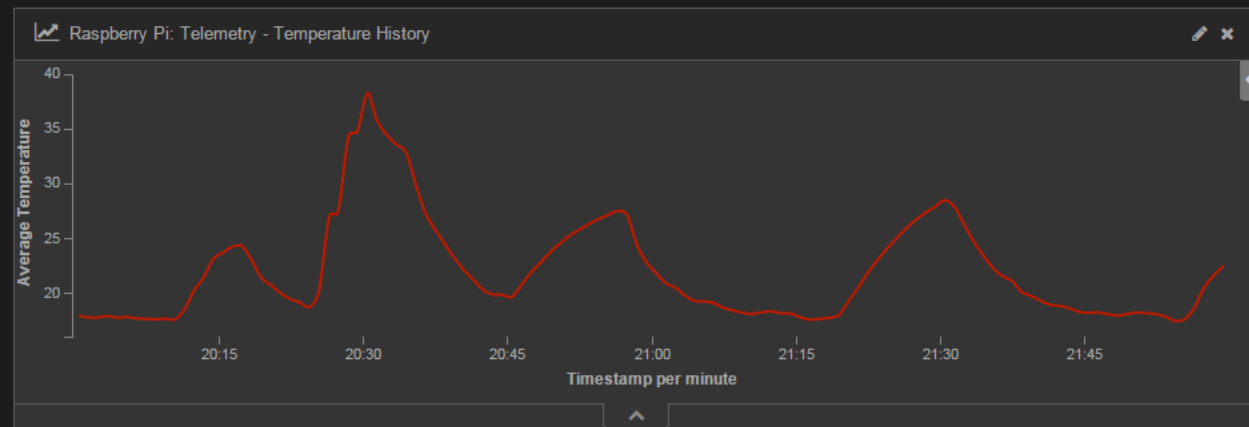
```
{rabbitmq_auth_backend_ldap, [
    {servers, ["ldap_server_address"]},
    {use_ssl, false},
    {port, <ldap_port>},
    {log, true},
    {user_dn_pattern, "cn=${username},ou=o,dc=abc,dc=com"},
    {resource_access_query, ... },
    {tag_queries, ... }
]}
```

...]

Logs









Gracias

