

Guia de Integrações ServiceNow

Esta página descreve exclusivamente o fluxo de integrações com o ServiceNow dentro da Comunicação Transacional, desde a leitura do tópico *comunicacaoTransacional*, recuperação/criação de cliente (consumer/contact), abertura do Case, disparo do WhatsApp, atualização do Case conforme o resultado e publicação da saída no *comunicacaoTransacionalResponse*. Também cobre o consumo do *omnichannelRespostaCampanha* para atualizar o Case com leitura/retornos do cliente.

Etapas do Fluxo

Ingestão & Gatilho

- brom-ctx-functions* (FaaS) lê o tópico Kafka *comunicacaoTransacional*.

ServiceNow — Identificar Cliente e Abrir Case

- FaaS consulta o cliente no ServiceNow.
- Se não existir, cria o cliente.
- Abre o Case vinculado ao cliente, já com os campos de rastreo (ex.: *id_correlacao*, *num_pedido*).

WhatsApp — Disparo via Weni

- FaaS chama a API do WhatsApp (Weni) usando o *case.number* ou *sys_id* (*id_case*) no contexto.
- Se HTTP 200: atualiza o Case informando “enviado”.
- Se HTTP diferente de 200: atualiza o Case informando falha e motivo.

Publicação de Resultado

- FaaS publica o resultado consolidado no tópico *comunicacaoTransacionalResponse* (padrão *success/partial_success/failure* e “resultados” por canal).

Inbound Weni Atualizações no Case

- FaaS consome o tópico *omnichannelRespostaCampanha* (eventos de mensagem entregue/lida e retorno do cliente).
- Atualiza o Case com leitura/entrega e, em caso de retorno:
 - SIM: fechar o Case.
 - NÃO: manter aberto, priorizar/atribuir conforme regra.
- Publica os retornos de API (e efeitos no Case) sempre no *comunicacaoTransacionalResponse*.

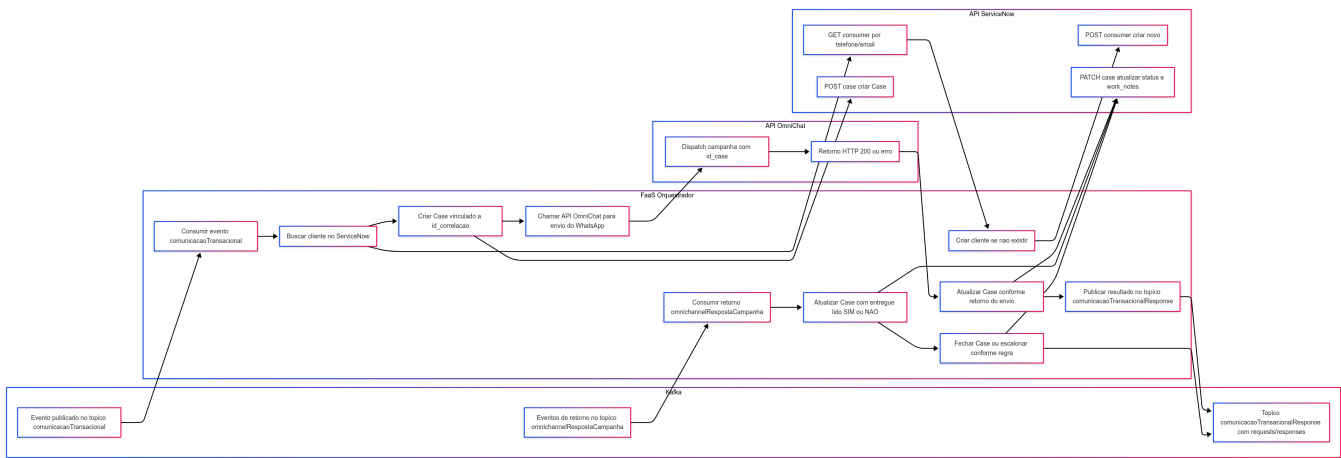
Componentes Principais

Componente	Descrição
brom-ctx-functions	FaaS de orquestração; integra ServiceNow e OmniChat; publica <i>comunicacaoTransacionalResponse</i>
ServiceNow API	ITSM para recuperar/criar cliente e abrir/atualizar cases
Weni API	Envio de WhatsApp (campanhas) e origem dos webhooks de status/retorno

Tópicos Kafka (envolvidos neste fluxo)

Tópico	Finalidade
<i>comunicacaoTransacional</i>	Entrada de eventos para iniciar a comunicação e abrir Case no ServiceNow
<i>comunicacaoTransacionalResponse</i>	Saída consolidada com requests/responses de APIs (ServiceNow/Whats) por canal
<i>omnichannelRespostaCampanha</i>	Eventos inbound do WhatsApp (entregue/lida/retorno do cliente) que atualizam o Case

Desenho do Fluxo



Fluxo — Chamadas de API e Padrões

1) Ler do tópico *comunicacaoTransacional*

Exemplo de payload de entrada (campos mínimos para ServiceNow + Whats):

```
```json
{
 "tims_evento": "2025-09-30T15:30:00Z",
 "cod_canais": ["whatsapp", "itsm"],
 "cod_jornada": "pedido_entregue_amanha",
 "cliente": {
 "id": "UUID",
 "nome": "João",
 "sobrenome": "Silva",
 "email": "joao@...",
 "num_telefone": "+5511999999999"
 },
 "dados_jornada": {
 "id": "OBM3028874-01",
 "num_pedido": "OBM3028874-01",
 "num_nota": 12345566,
 "remessa": "805566999",
 "dat_emissao": "2025-07-07",
 "url_nota": "https://.../nota/12345566"
 },
 "dados_whats": {
 "id_campanha": "xpro"
 },
 "dados_canal_itsm": {
 // a definir
 }
}
```
```

2) ServiceNow — Buscar cliente (csm_consumer)

GET consumer por cpf/cnpj, email, telefone (nessa ordem).

```
GET https://obramax.service-now.com/api/now/table/csm_consumer?sysparm_query=phoneLIKE{{num_telefone}}
^OReemailLIKE{{email}}&sysparm_limit=1
```

Se não encontrar, utilizar csm_consumer genérico CTGuest - ID xOeTBxFz98JIPlG00LagPt7aH4BSvmY9t3zy7CJ2cSk=

3) ServiceNow – Criar Case

POST https://obramax.service-now.com/api/now/table/sn_customerservice_case

```
```json
{
 "consumer": "<<<SYS_ID_DO_CONSUMER>>>",
 "short_description": "Confirmação de entrega do pedido OBM1020304-01 para 02/10/2025",
 "description": "Olá, <<<NOME_DO_CLIENTE>>>! Informamos que sua entrega está prevista para 02/10/2025. Responda SIM para confirmar, ou NÃO para remarcar/falar com um atendente.",
 "contact_type": "phone",
 "category": 0, // 0=Question, 1=Issue, 2=Feature
 "subcategory": 0, // 0=Question
 "priority": 3, // 1=Critical, 2=High, 3=Moderate, 4=Low
 "impact": 3, // 1=High, 2=Medium, 3=Low
 "urgency": 3, // 1=High, 2=Medium, 3=Low
 "assignment_group": "<<<SYS_ID_GRUPO_ATENDIMENTO>>>",
 "state": 10, // 10=Open (aguardando resposta do cliente)
 "opened_by": "<<<SYS_ID_USUARIO_CHAMADOR>>>"
}
```
```

Resposta:

```
```json
{
 "result": {
 "sys_id": "0397c743db43...",
 "number": "CS0001005"
 }
}
```
```

4) WhatsApp — Disparo via Weni?

Endpoint:

POST https://campaign-api.omni.chat/v1/campaigns/{ID_CAMPANHA}/dispatch?keepValidContacts=false

Payload (inclui id_case do ServiceNow dentro de context para rastreio):

```
```json
{
 "contacts": [
 {
 "context": {
 "cod_jornada": "pedido_entregue_amanha",
 "id_contexto": "pedido_entregue_amanha|UUID-cliente|805566999OBM3028874-01",
 "id_case": "CS0001005"
 },
 "fullNumber": "+5511999999999",
 "num_pedido": "OBM3028874-01",
 "remessa": "805566999"
 }
]
}
```
```

5) ServiceNow — Atualizar Case conforme retorno do Whats

Se HTTP 200 (enviado):

PATCH https://obramax.service-now.com/api/now/table/sn_customerservice_case/{{sys_id}}

```
```json
{
 "work_notes": "WhatsApp enviado com sucesso {{timestamp_envio}}"
}
```
```

Se HTTP != 200 (falha):

PATCH https://obramax.service-now.com/api/now/table/sn_customerservice_case/{{sys_id}}

```
```json
{
 "work_notes": "Falha no envio do WhatsApp: {{codigo_status}} {{erro}}"
}
```
```

6) Publicar saída no *comunicacaoTransacionalResponse*

```

```json
{
 "id_correlacao": "pedido_entregue_amanha|550e8400-e29b-41d4-a716-446655440000|805566999OBM3028874-01",
 "cod_canaais": [
 "whatsapp",
 "itsm"
],
 "response_status": "success",
 "resultados": {
 "whatsapp": {
 "status": "success",
 "request": {
 "contacts": [
 {
 "context": {
 "cod_jornada": "pedido_entregue_amanha",
 "id_contexto": "pedido_entregue_amanha|550e8400-e29b-41d4-a716-446655440000|805566999OBM3028874-01",
 "id_case": "CS0001005"
 },
 "fullNumber": "+5511999999999",
 "num_pedido": "OBM3028874-01",
 "remessa": "805566999"
 }
]
 },
 "response": {
 "status": 200,
 "message": "accepted"
 }
 },
 "itsm": {
 "status": "success",
 "request": {
 "contact_type": "social",
 "short_description": "Confirmação de entrega D+1 - WhatsApp",
 "priority": "4",
 "category": "entrega",
 "subcategory": "solicitacao",
 "consumer": "{{consumer_sys_id}}",
 "u_id_correlacao": "{{id_correlacao}}",
 "u_num_pedido": "{{num_pedido}}",
 "u_msisdn": "{{num_telefone}}"
 },
 "response": {
 "sys_id": "0397c743db43...",
 "number": "CS0001005"
 }
 }
 }
}
```

```

Observações:

- "response_status" deve refletir o agregado dos canais: *success*, *partial_success* ou *failure*.
- Sempre registrar o *request* e *response* de cada canal envolvido.

Consumo do *omnichannelRespostaCampanha* e Atualizações no Case

Eventos esperados

- Mensagem entregue/lida (atualizar timestamps no Case).
- Retorno do cliente (SIM/NÃO) contendo *case.number* (ou *id_case*) no payload.

Atualizações no Case

- Mensagem entregue/lida:

PATCH https://obramax.service-now.com/api/now/table/sn_customerservice_case/{{sys_id}}

```
```json
{
 "work_notes": "Mensagem entregue as {{ts_delivered}} e lida as {{ts_read}}"
}
```
```

Retorno SIM: fechar o Case (validar horário, se dentro fechar case, contrário?)

PATCH https://obramax.service-now.com/api/now/table/sn_customerservice_case/{{sys_id}}

```
```json
{
 "state": 6, // 6=Resolved
 "close_notes": "Cliente confirmou a entrega para 02/10/2025. Caso encerrado.",
 "work_notes": "Confirmação recebida do cliente: SIM."
}
```
```

Retorno NÃO: manter aberto, priorizar/atribuir conforme regra

PATCH https://<instance>.service-now.com/api/now/table/sn_customerservice_case/{{sys_id}}

```
```json
{
 "state": 10, // Open
 "assignment_group": "<<<SYS_ID_GRUPO_ATENDIMENTO_HUMANO>>>",
 "priority": 2, // dar mais urgência?
 "work_notes": "Cliente respondeu NÃO. Encaminhado para atendimento humano para remarcar a entrega."
}
```
```

Publicar todos os retornos no *comunicacaoTransacionalResponse*

- Cada evento processado (entrega/leitura/retorno SIM/NÃO) deve gerar uma entrada no *comunicacaoTransacionalResponse* com request /response (quando couber) e status por canal.

Erros, Retentativas e Observabilidade

- ServiceNow/OmniChat HTTP != 200: registrar *work_notes* com erro e publicar no *comunicacaoTransacionalResponse*; aplicar retry/backoff no FaaS.
- Timeouts: tratar como *failure* do canal, publicar no *comunicacaoTransacionalResponse* com o erro retornado (ex.: 504).
- Logging estruturado: incluir *id_correlacao*, *case.number/sys_id*, *message_id* do Whats e timestamps de cada marco.

Referência ServiceNow (REST): <https://developer.servicenow.com/dev.do#!/reference/api/yokohama/rest>