Document Navigation

ZephyrCart

ERP bridge

Order data flows from ZephyrCart into your ERP (Microsoft Dynamics 365 Business Central or SAP Business One in 90 % of the deployments we see). The recommended shape is a thin worker that subscribes to `order.created` and `order.refunded` webhooks and translates them into ERP API calls.

Reference architecture

```
ZephyrCart #### webhook ####  Bridge worker  #### ERP API #####  Dynamics /
SAP
                                (your code)
                                    #
                                    #
                             Dead-letter queue
                             (for replays)
```

Why a worker, not a direct webhook to the ERP

Three reasons:

1. ERPs do not authenticate webhooks the same way we sign them. The worker terminates the signature, translates, and reauthenticates with the ERP's own scheme.
2. Field mapping changes more often than the ZephyrCart event schema. Keeping it in your worker lets you ship mappings without touching either side.
3. ERPs go down for maintenance windows. The worker is where you buffer.

Recommended technology

- A small Cloud Run / AWS Lambda / Azure Function — request load is exactly "one execution per order event," which is the cheapest serverless shape there is.
- A persistent queue between webhook receipt and ERP push (Cloud Tasks, SQS, Service Bus) so the ERP outage doesn't lose data.

Idempotency at the ERP

Send the ZephyrCart `order.id` as the external reference on the ERP record. If the ERP returns "already exists," consider that a success and ack the webhook.

A worked example

A Node.js worker on Google Cloud Run, posting into Dynamics 365 Business Central:

```
import { verifyZephyrSignature } from "./signing";

export async function handle(req: Request): Promise<Response> {
  const body = await req.text();
  if (!verifyZephyrSignature(body, req.headers.get("x-zephyrcart-
signature")!)) {
    return new Response("bad signature", { status: 401 });
  }
  const event = JSON.parse(body);
  if (event.type !== "order.created") {
    return new Response("ignored", { status: 200 });
  }

  await dynamicsClient.createSalesOrder({
    externalReference: event.data.id,                // idempotency anchor
    customerEmail: event.data.customer.email,
    currency: event.data.currency,
    lines: event.data.line_items.map(toErpLine),
  });

  return new Response("ok", { status: 200 });
}
```

Failure modes worth handling

Failure	How to recover
ERP returns 5xx	Return 5xx from the worker — ZephyrCart retries
ERP returns 4xx because the SKU is unknown	Log to dead-letter queue, page the catalog team
Worker times out at 10 seconds	Ack early, push to your own queue, process async
Signature verification fails	Return 401; investigate clock skew first