The right box lights up.

Merkora puts e-paper labels with an RGB LED and a button on kitting boxes, SMD magazines and shelves. Your ERP decides what each label shows and when it blinks, so nobody has to search the racks.



Tile labels on kitting boxes, the middle one with its LED lit. Rendered from the design model; the boxes, shelf and label data are stand-ins.

Kitting boxes

Work order, due date and priority on the box. When planning moves the order up, the label turns red on its own.

SMD magazines

Which job a magazine is set up for. When the line needs it, it blinks green on the rack.

Warehouse bins

Part number and quantity to take. The light leads the picker to the bin; the button confirms the pick.

Kanban

A card that reorders itself. Press the button when the bin runs empty and your ERP gets the signal.

In development. The hardware is designed; nothing has been fabricated, tested on hardware or certified yet. The software runs against a bridge simulator until the first labels exist. Numbers in this brochure are design targets.

Two labels and an access point

Panels in the sizes the shelf-label industry makes by the million, so they stay affordable and easy to source. The housings are modelled in code, and every picture here is rendered from that model.



Strip, Tile and access point, rendered from the design model.

Strip 2.9″

For SMD magazines, bins and shelf edges.

Panel	296 × 128, black and white or four colours
Size	100 × 44 × 12 mm
Mount	Slides onto an adapter plate stuck to the magazine

Tile 4.2″

For kitting boxes, carts and stations.

Panel	400 × 300, black, white, red and yellow
Size	100 × 98 × 12 mm
Mount	Hooks over the box rim, sits in a card pocket, or sticks on

Access point

On the ceiling, wired to the bridge.

Power	Power over Ethernet, 802.3af
Size	About 130 × 90 × 35 mm, plus the dipole
Mount	Strut channel, ceiling or wall

What every label has

The LED colours the API asks for by name:

- ○ Locate: white
- ● Pick: green
- ● Attention: yellow
- ● Alert: red

Every value here is a design target. Measured numbers follow once labels run in production.

LED	RGB, designed to start blinking within about 2 s of a request
Button	One per label; presses are reported back to your ERP
Battery	Two CR2450 coin cells behind a snap door that a coin releases; designed for 5+ years at a few updates a day
Radio	Bluetooth 5.4 Electronic Shelf Label profile, encrypted
Pairing	Tap with a phone over NFC, or scan the QR code
Housing	Static-dissipative plastic, for use next to ESD-sensitive parts

From your ERP to the shelf

One request travels through five stations. The light comes on before the label has finished redrawing.

1 **Your ERP**
Changes an order or asks for a box to be found. ERPNext has an app; any other system uses the REST API.

2 **Control center**
Works out what every label should show. Runs in our cloud or on your own servers.

3 **Bridge**
A small container in your network. It draws the label images and talks to the access points.

4 **Access points**
On the ceiling, powered over Ethernet. Three or four cover about 1,500 m².

5 **Labels**
Redraw and blink, using the Bluetooth 5.4 standard for shelf labels.

Bridge

One container per site, next to the access points.

- Renders templates into panel images and sends them to the labels through the access points.
- Runs the MQTT broker the access points connect to, with a login per access point.
- Keeps a queue and the desired state on disk, so labels keep updating and LEDs keep answering while the control center is out of reach.
- Needs only outbound HTTPS. The control center reaches it through a tunnel, a VPN or your local network.

Control center

The single place that knows what every label should show.

- Sites, bridges, access points, labels, templates and their bindings to ERP records.
- Templates are HTML with placeholders; the same renderer draws the previews and the panels.
- Reads ERP data over its REST API and redraws only the labels whose content changed.
- A web interface for setup and an event log of button presses, redraws and battery levels.
- Postgres with row-level security per organisation; secrets in a vault, API keys stored as hashes.

Run it where you want

Every setup uses the same containers. Nothing is tied to our cloud, and you can move between modes later.

Setup	Control center	Bridge	How they connect
Merkora Cloud	Hosted by us	In your network	A tunnel to the bridge, for example Cloudflare Tunnel
Your cloud	Your Supabase project and one container	In your network	Tunnel, VPN or an open port
On premises	A server in your network	Next to it	Your local network, nothing else

Setting up a label

1. Tap the label with a phone over NFC, or scan its QR code. That gives its Bluetooth address and model.
2. The control center registers it; the bridge picks the access point that hears it best and pairs it with encryption keys.
3. Bind the label to a record, such as a kitting box, and choose a template.
4. From then on it follows the record. For a battery change it comes off its plate; the plate stays on the box.

What to expect

LED	Blinks within 2 s of a request
Strip, BW	New image within 10 s
Tile, colour	New image within 40 s, about 20 s of it the redraw
Battery	5 years or more at 4 updates and 10 locate requests a day
Coverage	3–4 access points for about 1,500 m² with racks

Design targets, to be measured on the first labels.

Your ERP stays in charge

Merkora doesn't keep its own copy of your orders. It asks your ERP what a label should show and tells it when someone presses a button.

An app for ERPNext

For Frappe and ERPNext v15 and v16.

Locate buttons

Work orders, warehouses and kitting boxes get a Locate button that makes every bound label blink.

Labels follow your records

A saved document sends a short signed hint. Merkora re-reads the record over the REST API, so lost or repeated hints do no harm.

Presses come back

A button press adds a comment to the record's timeline; a low battery notifies a role you choose.

A REST API for everything else

Any ERP, MES or WMS can use it, with webhooks for the way back.

Make every label on kitting box KIT 164 blink:

```
curl -X POST https://your-server/functions/v1/api/v1/locate \
  -H "Authorization: Bearer mk_live_…" \
  -H "Content-Type: application/json" \
  -d '{
    "entity": { "doctype": "Warehouse", "name": "KIT 164" },
    "preset": "locate"
  }'
```

Button presses, finished redraws and low batteries arrive as signed webhooks.

Security

- Labels only accept encrypted commands (Bluetooth ESL profile encryption, keys set at pairing).
- Calls to the bridge and hints from the ERP are signed with HMAC-SHA256 and refused after 300 s; each bridge has its own token.
- Bridge tokens and API keys are stored as hashes and shown once.
- On premises, nothing leaves your network.

Open by design

- Software and firmware will be published under Apache-2.0, the hardware under CERN-OHL-W-2.0, once the first version is stable.
- Electronics in KiCad, housings in build123d: free tools, so anyone can open and check the design files.
- Labels use the Bluetooth ESL standard, not a proprietary radio.

Made by people who build electronics

Newmatik assembles circuit boards for other companies. We started Merkora because we needed it on our own floor: kitting boxes whose orders change every day, and SMD magazines that have to reach the right line at the right time. We design the boards ourselves, will build them on our own assembly lines, and run every version in our own production before anyone else gets it.

Want labels on your floor?

We are looking for a few manufacturers to run pilot sets with us. Tell us what you would label and roughly how many.

service@newmatik.com

merkora.dev