

The Odoo AI Feature Map

Every AI capability that actually ships in Odoo 20.0 Community: the module it lives in, the endpoint it calls, and what leaves your database when you use it. Read from a running instance, not from a release note – and compared, line by line, with 19.0.

TESTED ON	COMPARED WITH	METHOD	LAST UPDATED
Odoo 20.0 Community	Odoo 19.0 Community (edition 1)	source read + shell probe, clean database	21 September 2026

WHAT CHANGED SINCE EDITION 1

Odoo 20 moved the AI around more than it added any

Edition 1 described 19.0. We re-read the 20.0 source and re-ran the probe on a clean 20.0 Community database. Four things moved:

- **The free-form ChatGPT box is gone from the Community editor.** The dialog that let anyone write a prompt in any rich-text field no longer ships in `html_editor`. The server route that served it is still there, and other features now call it.
- **Translation changed provider.** The translate button in the editor and in Discuss now goes to `ots.api.odoo.com`, labelled Google Translate. ChatGPT stays available as a second translator only in debug mode. Unlike the generation endpoint, this address has no system parameter to override it.
- **The website configurator asks the model six times, not once.** It now also suggests themes, a colour palette, hero titles and a positioning, and matches an industry you type freely against Odoo's list.
- **API keys gained a scope.** You choose it when you create the key. Community offers a single scope, `rpc`, and the key-name field suggests « Odoo MCP » as an example.

Two corrections to edition 1 as well, stated rather than patched: it missed two features that were already in 19.0 (website page translation and eCommerce category generation), and it said CRM sets the lead-scoring start date to the installation date. It sets it **eight days before** the installation date, in 19.0 as in 20.0.

Sorted by what leaves your database, not by what sounds impressive

Most write-ups of Odoo's AI features list what the buttons do. That is the least interesting question, because the buttons are easy to try. The question that decides whether a feature is usable in a regulated company, or in any company with a data protection officer, is where the text goes when you press the button.

So every capability here carries an **egress** value: nothing leaves, something leaves, or customer data leaves. Of the six capabilities that ship in 20.0 Community, four send content to Odoo's servers, one sends contact data to be matched, and one sends nothing at all – still the one nobody writes about.

Nothing on this map is quoted from marketing material. Each row was read in the source and then confirmed against a running instance with the probe script at the end, which you can run against your own database in about a minute.

THE MAP

Six capabilities ship in 20.0 Community. One left. The agents are not here.

CAPABILITY	MODULE	WHAT IT CALLS	EGRESS	BILLING	20.0 VS 19.0
Translation in editor and Discuss	html_editor, mail	ots.api.odoo.com	SELECTED TEXT + DB UUID	Server-side quota	Changed provider
Website configurator	website	olg.api.odoo.com	SITE CONTEXT	Server-side quota	Expanded
Website page translation	website	olg.api.odoo.com	PAGE TEXT	Server-side quota	Unchanged*
eCommerce category generation	website_sale	olg.api.odoo.com	INDUSTRY + LANGUAGE	Server-side quota	Unchanged*
Predictive lead scoring	crm	nothing – local	NONE	Free	Unchanged
Lead enrichment & partner autocomplete	crm_iap_enrich, partner_autocomplete	IAP (reveal)	CONTACT DATA	IAP credits	Unchanged
Free-form text generation in the editor	html_editor	–	FEATURE GONE	–	Removed

CAPABILITY	MODULE	WHAT IT CALLS	EGRESS	BILLING	20.0 VS 19.0
Configurable AI agents	—	—	UNKNOWN	—	Not in Community
MCPserver for external assistants	—	—	UNKNOWN	—	Not in Community
Voice dictation	—	—	UNKNOWN	—	Not in Community
Record cards in Live Chat	—	—	UNKNOWN	—	Not in Community

* Already in 19.0, and missing from edition 1.

THE SIX, AND THE ONE THAT LEFT

What each one is, and what it costs you in data

Translation in the editor and in Discuss		EGRESS: SELECTED TEXT + DB UUID
WHERE IT LIVES	A plugin, TranslatePlugin, registered both in the editor's plugin set and in Discuss's. The second registration is still the one people miss: the translate button is in your internal messaging too.	
WHAT CHANGED	In 19.0 the plugin was ChatGPTTranslatePlugin and went through the generation endpoint. In 20.0 the dialog opens on a translator labelled Google Translate, which posts to <code>ots.api.odoo.com/api/html_editor_translate/1/google_translate</code> . ChatGPT is added as a second choice only when debug mode is on.	
WHAT LEAVES	The selected text, the target language, and <code>database.uuid</code> . In Discuss, the selected text is a colleague's message — which may be about a customer, a price, or a salary. What Odoo's server does with it next is not visible from the client code; the dialog calls it Google Translate.	
WORTH KNOWING	The address is a constant in <code>html_editor/controllers/main.py</code> . There is no system parameter to point it elsewhere, where the generation endpoint has one. Blocking it at the firewall is the only switch left to an administrator.	

Website configurator EGRESS: SITE CONTEXT	
WHERE IT LIVES	website, in the configurator that runs when you create a site.
WHAT IT DOES IN 20.0	Six calls where 19.0 made one. It still writes the placeholder text of the generated pages for your industry. It now also picks the best-fitting themes, recommends a colour palette with a one-line reason, proposes six hero titles and six positionings, and maps an industry you typed freely onto Odoo's list, with a confidence score.
WHAT IT CALLS	<code>/api/olg/1/generate_placeholder</code> and <code>/api/olg/1/chat</code> on <code>olg.api.odoo.com</code> , overridable through <code>website.olg_api_endpoint</code> ; plus the editor's own <code>/html_editor/generate_text</code> route for the browser-side calls.
WHAT LEAVES	Your industry, website type, positioning, language, and Odoo's list of themes or palettes. No customer data – the database is new at that point.
WORTH KNOWING	Text generated here lands in your pages as ordinary content. If your policy requires disclosing machine-written copy, the disclosure has to be added by hand: Odoo keeps no marker of what was generated.

Website page translation EGRESS: PAGE TEXT	
WHERE IT LIVES	website, in the translation option of the website builder. It was already in 19.0; edition 1 missed it.
WHAT LEAVES	The text of the page, cut into chunks and sent through <code>/html_editor/generate_text</code> with a translation instruction. Unlike the translate button above, this one stayed on the language model.

eCommerce category generation EGRESS: INDUSTRY + LANGUAGE	
WHERE IT LIVES	<code>website_sale</code> , when the configurator sets up a shop. Also already in 19.0, also missed by edition 1.
WHAT IT DOES	Asks <code>/api/olg/1/chat</code> for eight product categories with a short persuasive description each, for your industry and language. If the answer is not valid JSON, or the quota is reached, it logs a warning and carries on without them.

Predictive lead scoring EGRESS: NONE	
WHERE IT LIVES	crm, in Community. No subscription, no credits, no external service. Unchanged in 20.0.
WHAT IT IS	A naive Bayes classifier. Every won or lost opportunity increments a frequency table, <code>crm.lead.scoring.frequency</code> , and the probability field on each lead is computed from it. This is the only machine learning that Odoo actually runs on your own hardware.
IT IS ALREADY ON	Installing CRM sets <code>crm.pls_start_date</code> to eight days before the installation date (<code>datetime.now() - timedelta(days=8)</code> in <code>crm_lead_prediction_data.xml</code> ; edition 1 said « the installation date », which was wrong) and <code>crm.pls_fields</code> to seven fields: phone state, email state, state, country, source, language, tags. Scoring accumulates from day one, whether or not anyone decided to use it.
WHAT LEAVES	Nothing. The classifier itself never leaves the database.
THE CATCH	It learns from your history, so it inherits your history's biases. Country and language are scoring fields by default, which is worth a deliberate decision rather than an inherited one.

Lead enrichment and partner autocomplete EGRESS: CONTACT DATA	
WHERE IT LIVES	<code>crm_iap_enrich</code> , <code>crm_iap_mine</code> and <code>partner_autocomplete</code> , over Odoo's IAP platform. Unchanged in 20.0.
WHAT LEAVES	An email domain or a company name, sent to Odoo's servers to be matched against a third-party database. Unlike the generation features, this one is billed: the services <code>reveal</code> and <code>partner_autocomplete</code> appear in <code>iap.service</code> and draw on credits.
WHY IT IS ON THIS MAP	It is not a language model, and calling it AI is generous. It is here because it is the feature most often demonstrated as AI, and because it is the one that moves real contact data.

Free-form text generation in the editor — removed		GONE FROM COMMUNITY
IN 19.0	A ChatGPT dialog in the editor toolbar: write a prompt, get a draft, ask for alternatives, insert. It sent your prompt, the conversation history and <code>database.uuid</code> to <code>olg.api.odoo.com</code> .	
IN 20.0	The dialog's files are no longer in <code>html_editor</code> , and nothing else in Community opens a free prompt. The route <code>/html_editor/generate_text</code> survives, with the same quota (<code>limit_call_reached</code>) and the same override (<code>html_editor.olg_api_endpoint</code>): the configurator and the translators above use it.	
WHAT IT MEANS	A Community user can no longer type a prompt into a rich-text field. We have not run Enterprise 20.0, so we do not say where, or whether, the dialog went.	

WHAT IS NOT IN COMMUNITY 20.0

The sockets ship. The agents plug in from elsewhere.

Configurable AI agents, an MCP server, voice dictation and record cards in Live Chat are announced by Odoo. **None of them is in the 20.0 Community source**, so on the database we probed they do not exist. What Community 20.0 does carry is the place where they attach:

- API keys now take a **scope**, chosen at creation. Community declares one, `rpc`. The name field's example is « Odoo MCP »: this is the key an external assistant logs in with, and anything it can then read is what that user can read.
- The Live Chat operator priority names `ai_agent` next to `chatbot_script` in a code comment. The agent operator itself is not in Community.

We could paraphrase the announcements. Every other write-up already does, and that is exactly the habit this map exists to break. These rows get filled in when we have run them on an instance we control.

CHECK IT YOURSELF

The probe, and its output on our instance

A reference you cannot check is a reference you have to trust. Run this against your own database – it reads, never writes, and makes no external call. The same script runs on 19.0 and 20.0 (Odoo 20 removed `get_param()`, which broke the edition 1 probe on 20.0):

```
odoo-bin shell -c /etc/odoo.conf -d YOUR_DB --no-http < probe.py > out.txt 2>&1
```

Odoo's shell writes on stderr, which is why both streams are redirected. Our own run on a clean 20.0 Community database, verbatim:

```
=== build =====
version                20.0
database               vs20_oais_featuremap

=== modules =====
html_editor            installed
website                installed
website_sale           installed
crm                    installed
mail                   installed
iap                    installed
partner_autocomplete  installed
crm_iap_enrich          installed
crm_iap_mine           installed

=== text generation =====
html_editor.olg_api_endpoint  (unset -> default)
website.olg_api_endpoint      (unset -> default)
database.uuid is set          True
editor ChatGPT dialog ships   False

=== translation =====
translate via Google (ots)     True
translate via ChatGPT, by default False

=== IAP services declared =====
reveal                      Lead Generation
partner_autocomplete        Partner Autocomplete
sms                          SMS

=== API keys =====
keys in this database         0
selectable scopes            rpc

=== predictive lead scoring =====
frequency rows                0
crm.pls_start_date            2026-09-13
crm.pls_fields                 phone_state,email_state,state_id,country_id,source_id,lang_id,tag_ids
crm.lead has probability       True

=== end =====
```

The IAP block is still the one to read twice. What matters there is the absence: none of the declared services is text generation or translation. That is how you know these features are not billed to your credits – and, at the same time, that their limit is set by someone other than you. (Edition 1 listed a fourth service, snailmail: that module was installed on that database, not on this one.)

WHAT THIS MAP IS NOT

Not a review, and not a buying guide

It does not tell you whether Odoo's AI is good. It tells you what exists, where it lives, and what it sends. Those are the facts a decision rests on; the judgement is yours and it depends on your industry, your regulator, and your data.

The map is re-run at every Odoo release. If this was useful, the deeper work – configuration, guardrails, and what breaks in production – is what we are writing about at odooaiskills.com.

The Odoo AI Feature Map, edition 2, 21 September 2026. Verified on Odoo 20.0 Community; edition 1 (16 August 2026) covered 19.0. Published by Odoo & AI Skills LLC, 8206 Louisiana Blvd NE, Ste A #9881, Albuquerque, NM 87113, United States.

Odoo is a trademark of Odoo S.A. This publication is independent and is not affiliated with, endorsed by, or sponsored by Odoo S.A.

Share it freely, unmodified and with attribution.