

Tetrade Agent Router Enterprise: Smart Routing

Send every request to the right model. Pick it yourself, write rules, or let Tetrade-auto choose. Smart switching cut the cost per finished agent task in half.

EXECUTIVE SUMMARY

Details on page 3 →



50%

lower cost per finished task



Same

share of tasks finished as the strong model



24%

shorter runs (2.0 vs 2.7 min)

Cost tiering

Match model price to the value of each request. Work that can wait runs on cheaper models.

Model lifecycle

Update, replace, and retire models behind one stable name. Apps need no new release.

Tenancy tiering

Route by customer, plan, or environment. Each tenant gets its own model policy.

Smart Routing Covers the Full Range of Control with 3 Ways

DETERMINISTIC

PROBABILISTIC

YOU CHOOSE

Model choice

User → Approved catalog → Chosen model
Admins set each team's catalog. Users pick from it.

```
team:      Marketing
approved: small, standard
pick:      standard
```

YOUR RULES CHOOSE

Attribute-based routing

Request + tag → Ordered rules → Model
Admins set the traffic classes. App owners choose within them.

```
trial agent → small
support agent → frontier
default → standard
```

TETRDE-AUTO CHOOSES

Tetrade-auto

Request + context → Decision model → Best fit
A Tetrade decision model picks for each request, inside admin policy.

```
context: code review
policy: dev, max models
picked: frontier
```

How a Request Is Routed

01

Request arrives

The app sends a model name, tags, or both.

02

One policy applies

The router finds the one policy for this API key.

03

A model is chosen, in one of three ways

CHOICE The asked-for model must be in the catalog.

RULES Rules run top to bottom. The first match wins.

AUTO The decision model picks the best fit within policy.

04

A model serves it

If nothing matches, the default model serves it.

05

The decision is saved

The record shows the mode, the reason, and the model.

Common Use Cases

DETERMINISTIC

PROBABILISTIC

YOU CHOOSE

Model choice

- 01 Give each team an approved list**
Admins assign models to each team. Users pick only from that list.
- 02 Hold a group to cheaper models**
Create a project with only low-cost models. Issue the group's keys there. Nobody configures anything.
- 03 Stop hardcoding model names**
Publish one name for apps to call. Repoint it when a model changes. No redeploy.

YOUR RULES CHOOSE

Attribute-based routing

- 04 Make overnight work cheaper**
Tag batch jobs and route them to a cheap model. The nightly job adds one header.
- 05 Serve tiers different models**
Tag each request with the plan. Order rules by value.
- 06 Plan big, execute cheap**
An agent plans once, then takes many small steps. Tag each call by task. One run uses two models.
- 07 Test a model on one cohort**
The app tags beta users. The rule sends only them to the new model.
- 08 Retire a model gradually**
Move batch to the new model first. When it works, change the default.

TETRATE-AUTO CHOOSES

Tetrates-auto

- 09 Let the router pick for each request**
Apps call tetrates-auto. The decision model reads the request and picks a model.
- 10 Match the model to the task**
Simple requests go to small models. Hard work, like code review, goes to a frontier model.
- 11 Keep every pick inside policy**
Admins set the limits, such as max models for dev. Tetrates-auto only picks within them.

EXAMPLE OF POTENTIAL IMPACT

Smart switching cut the cost per finished agent task in half

A decision model chose when an agent could hand work to a cheaper model. The agent finished the same share of tasks at half the cost.

50% lower cost per finished task

Same tasks finished as the strong model alone

24% shorter runs (2.0 min vs 2.7 min)

HOW SMART SWITCHING WORKS

A Strong model **B** Cheap model

Agent task

A Strong model starts

Y Decision model checks at each work step

B Cheap model does routine work

A Strong model does a short final check

A A A B B B B B B B B B B A

One real run: strong model starts, cheap model does the bulk, strong model checks.

FOUR ROUTING RULES COMPARED

Cost: strong model only = 100%

Rule and call pattern

A A A

Strong model only

Tasks finished

Cost per finished task

Time

5/10 **100%** 2.7 min

A B A

Smart switch (decision model) · timed

6/10 **50%** 2.0 min

A B A

Cheap after step 1, strong check · fixed

4/10 **13%** 1.9 min

A B B

Cheap after step 1, no check · fixed

3/10 **17%** 2.0 min



Blind handoffs cost less but finish 20% to 40% fewer tasks. Choosing **WHICH** calls to move is what saves money.

What this could mean for a company spending \$1M a year on agent calls

ILLUSTRATIVE EXAMPLE

Strong model only		\$1,000,000
Smart switch with a decision model	Potential saving ~\$500,000 a year	~\$500,000
Cost of the decision model		<\$1,000

Smart switch figure is for the same finished work. Decision model: 12 million decisions cost about \$322 at list price.

Blind handoff: \$130K to \$170K, but 20% to 40% more failed tasks that need reruns or people.

HOW AGENT ROUTER ENTERPRISE APPLIES THIS

Shows use cases 06 and 10 from page 2.



Tetratrate-auto runs the decision model



Every pick stays inside admin policy



Every handoff is saved in the audit record