How a customer is matched to a tribe

The three steps behind your location finder, your chatbot and API lead routing — and what to change when a customer is sent to the wrong tribe.

Several parts of Gorilla Dash have to answer the same question: given where this person is, which of your tribes is theirs? (A **tribe** is one location, branch or business unit inside your organisation.) The location finder on your website asks it, the website chatbot asks it when a visitor says where they are, and the API asks it both when a store locator calls the **tribe search endpoint** and when an enquiry arrives with an address on it. They all use the same three-step search, so learning it once explains all of them — and tells you which setting to change when an answer is wrong.

It always starts with a point on the map

Nothing can be matched until the customer's location is a single latitude and longitude. If coordinates were supplied — by a visitor who allowed their browser to share their position, for example — those are used as the starting point, and the address is not looked up. Otherwise the address, suburb or postcode they typed is looked up first, and whatever point that produces becomes the starting point for everything that follows.

Either way the search ends up with a postcode as well as a point, because step 2 below needs one. When coordinates were supplied on their own, the postcode is worked out by looking the coordinates back up, so a visitor who shared their position is still matched against your territory postcode lists. And if a visitor refuses to share their position, some browsers send a pair of zeroes rather than nothing at all; those are discarded and the typed address is used instead, so nobody is matched to a point in the middle of the ocean.

> **WHY THIS MATTERS MORE THAN IT SOUNDS**
> A place name always becomes exactly one point, however large the place is. "Sydney" is a single point near the middle of Sydney, not the whole of it. That one fact explains most surprising results, and it is why a city name and a street address behave so differently.

The three steps

Step	What is checked
1. Drawn territory	Tribes whose drawn boundary contains the point.
2. Territory postcodes	Tribes whose postcode list contains the postcode the address produced.
3. Distance	The tribes physically closest to the point, in a straight line.

Steps 1 and 2 are treated as one group: any tribe found by either is a tribe whose territory covers this customer, and within that group the nearest is offered first. Only if that group does not produce enough answers does step 3 fill the rest of the list from the tribes that are simply nearby.

The practical rule that follows is worth remembering: **a tribe whose territory covers the address beats a nearer tribe whose territory does not.** That is the whole point of setting territories, and it is why a customer is sometimes sent past a closer tribe.

ONLY ACTIVE TRIBES WITH COORDINATES ARE SEARCHED

Every step needs a latitude and longitude on the tribe itself, and only tribes with a status of Active take part. A tribe that has never had its address placed on the map, or one still marked Testing or Opening Soon, is invisible to all three steps no matter how carefully its territory is drawn.

Where each answer is used

Where	What it does with the result
Location finder on your website	Shows the nearest several tribes and lets the customer choose.
Website chatbot	Ties the conversation, and any enquiry from it, to one tribe. See **How the chatbot works out which tribe a visitor means**.
The tribe search endpoint in the API	Returns the tribes in distance order for a store locator you have built yourself. See **Finding the nearest tribe: the tribe search endpoint**.
An enquiry pushed in through the API	Takes the single best match and assigns the enquiry to that tribe — but only if the enquiry did not name a tribe itself. See below.

The difference in the last row is worth noting. A location finder shows several tribes and lets a person decide, so a near miss costs nothing. An incoming enquiry has no one to ask, so the first answer becomes the tribe that gets the lead. Territories matter most where nobody is there to correct them.

AN ENQUIRY CAN NAME ITS OWN TRIBE, AND THAT WINS

Before any of the three steps run, an incoming API enquiry is checked for a tribe it has named for itself. If it was sent using a tribe's own API key, or carries a **tribe_slug** or a **tribe_name** that matches one of your tribes, that tribe takes the lead and the address is never looked at. An integration quietly sending the same value on every enquiry will make every territory change look as though it did nothing.

When the address cannot be looked up

Customers type "N/A", misspell their suburb, or enter a place that does not exist. Because the search cannot run without a point, these are handled deliberately rather than left to chance. A location finder falls back to a sensible stand-in tribe so the visitor sees something rather than an error, and an enquiry arriving with an unusable address is still recorded, with no tribe attached, because losing the lead would be worse. A tribe on a result is therefore not proof that the address was understood.

That stand-in is the one place the Active-only rule does not hold: it prefers an Active tribe, but will fall back to one marked Opening Soon rather than show the visitor nothing at all.

If your network trades in more than one country, there is one more rule worth knowing. When the chatbot can tell which country the visitor is in, it puts country ahead of status: a tribe in the visitor's own country is chosen before one anywhere else, even where the home-country tribe is only Opening Soon and the overseas tribe is Active. That is deliberate, because a visitor who cannot be placed on the map is at least shown somebody they could plausibly deal with. The location finder on your website and the tribe search in the API are not told the visitor's country, so in those two an Active tribe always comes first.

When a customer is sent to the wrong tribe

Work through these in order. Most reports come down to the first, the second or the last.

1. If the lead came in through the API, check whether it named a tribe itself. An enquiry sent with a tribe's own API key, or carrying a **tribe_slug** or **tribe_name**, never reaches the search at all.
2. Find the exact address the customer gave, and check whether the tribe that answered has a drawn boundary covering it. A boundary drawn generously years ago is the usual culprit.
3. Check the answering tribe's postcode list for the customer's postcode. A code pasted into the wrong tribe wins over distance and is easy to miss.
4. Check that the tribe you expected to answer is Active and has coordinates on its address. A tribe missing either cannot be returned at all.
5. Ask whether the customer named a whole city rather than a suburb or street. A city name resolves to its centre, so whoever's territory covers the centre answers for the entire city.

A CITY SPLIT BETWEEN FRANCHISEES NEEDS ITS BOUNDARIES REDRAWN

Where several tribes share one city, a customer who names only the city is always matched to whoever covers the middle of it, because that is the single point the city name produces. The chatbot is the exception: when a visitor names a city and more than one tribe sits within about 30 kilometres of it, it offers up to three nearby tribes for the visitor to pick from, which **How the chatbot works out which tribe a visitor means** explains. Everywhere else, the answer is to draw each tribe's boundary so that the suburbs it genuinely serves fall inside it — which is also the only fix when a suburb, rather than the whole city, is being matched to the wrong tribe.