

# The Business Case for Self-Hosted GPS Tracking

How to evaluate total cost, control, scalability, operational independence, and long-term value when choosing your GPS infrastructure model.

**Build the business case around value, responsibility, and evidence - not assumptions.**

# About This Guide

The Business Case for Self-Hosted GPS Tracking is an Open VTS publication for GPS and telematics business owners, founders, executives, CTOs, finance leaders, procurement teams, and infrastructure decision-makers evaluating the financial and strategic consequences of a tracking-platform operating model.

This guide is educational. It does not provide financial, legal, accounting, tax, security, or investment advice. Every cost model in this publication should be replaced with the reader's current supplier terms, infrastructure assumptions, labor costs, growth forecasts, and contractual conditions before a decision is made.

## Publishing Principle

**Do not ask whether self-hosting is always cheaper. Ask when the value of control, scale, flexibility, and operational independence exceeds the cost of operating the environment.**

## Product Scope Note

Open VTS is discussed as customer-controlled, self-hosted, proprietary vehicle tracking software. The core platform is not open source. The Open VTS mobile application is separately available as open-source software. Current pricing, licensing, update entitlements, support rights, and commercial conditions should always be verified against the latest official Open VTS terms.

## Copyright and Use

Copyright 2026 Open VTS. All rights reserved. This publication may be shared in its complete, unmodified form for evaluation and educational purposes. Commercial rights remain governed by the applicable Open VTS terms.

### OPENING THESIS

# The Business Case Is Bigger Than the Software Bill

A GPS platform can be inexpensive to buy and expensive to operate. It can also appear expensive at the beginning and become economically attractive as the operation grows. The invoice alone does not reveal the answer.

The business case sits across several layers: software, infrastructure, implementation, people, storage, integrations, support, growth, dependency, and the strategic value of controlling the environment in which the tracking operation runs.

**That means the correct question is not:**

**Which option has the lowest price today?**

**The stronger question is:**

**Which operating model creates the strongest long-term combination of cost, control, flexibility, continuity, and responsibility for our organization?**

This guide is designed to help you build that answer with your own assumptions.

#### EXECUTIVE SUMMARY

## A Business Case Should Measure Both Value and Burden

Self-hosting is sometimes presented as a simple financial argument: stop paying recurring SaaS fees and run the platform yourself. That framing is incomplete.

A self-hosted tracking operation may create economic advantages at some scales, but it also creates direct infrastructure and operating responsibilities. The organization may need to manage servers, networking, backups, monitoring, security, updates, and technical support processes. Those costs belong in the model.

At the same time, a cost-only comparison can miss strategic value. Control over infrastructure, data environment, integration architecture, device ecosystem, and upgrade timing may enable capabilities that do not appear as a line item on a software quote.

**This guide organizes the decision around six dimensions:**

- Cost Structure - what you pay to acquire and operate the platform
- Scale Economics - how cost changes as vehicles, data, and users grow
- Control Value - what infrastructure and data control are worth to the business
- Dependency & Continuity - where vendor or external dependency can affect operations
- Strategic Flexibility - what the organization can build, integrate, or change
- Responsibility Cost - what infrastructure ownership requires from people and processes

#### THE DECISION RULE

The business case is strongest when the measurable and strategic value of control exceeds the additional cost and responsibility required to operate it.

# Built for People Who Have to Defend the Decision

This guide is designed for organizations where vehicle tracking is part of the business model, a critical operational system, or a growing technology capability.

## GPS & Telematics Business Owners

For founders and operators deciding whether their platform economics, device ecosystem, customer operations, and dependence on a hosted vendor remain sustainable as the business grows.

## CEOs, CFOs & Finance Leaders

For decision-makers who need a 3-year or 5-year view of total cost, operating burden, break-even assumptions, and strategic value.

## CTOs & Technical Leaders

For teams translating architecture, integrations, data control, security, and scalability into business consequences.

## Fleet & Operations Leaders

For organizations assessing whether a different operating model could improve control, workflow flexibility, or long-term continuity.

## Procurement & Strategy Teams

For buyers comparing commercial structures without reducing the decision to price-per-vehicle alone.

The guide is particularly useful when the organization is growing, replacing an existing platform, launching a GPS service, considering self-hosted infrastructure, or preparing a formal business case for leadership approval.

## HOW TO USE THIS GUIDE

# Replace Every Example With Your Own Numbers

The models in this guide are designed to structure thinking, not to predict your result. Use them in five passes.

## 1. Establish the current state

Document what you pay today, which responsibilities the provider carries, what your fleet requires, and which constraints affect the business.

## 2. Build a comparable self-hosted model

Include software, infrastructure, implementation, labor, support, storage, security, backup, and change costs.

### 3. Model growth

Project the same business at realistic future fleet sizes and workloads. Avoid assuming that current usage remains constant.

### 4. Quantify what can be quantified

Use cost, time, labor, infrastructure, and recurring charges where reliable data exist. Keep non-financial benefits separate.

### 5. Test the assumptions

Run a proof of concept, validate infrastructure requirements, review commercial terms, and assign owners to operational responsibilities before presenting the final case.

#### **A useful discipline**

If a number came from a sales presentation, ask what assumption produced it. If a benefit cannot be quantified honestly, label it strategic value instead of inventing a dollar figure.

#### CONTENTS

## The Business Case in Three Parts

### PART I - THE ECONOMICS BEHIND THE HOSTING MODEL

- 01 - GPS Software Cost Is More Than the Price Tag
- 02 - SaaS and Self-Hosted Create Different Cost Structures
- 03 - Understand Total Cost of Ownership

### PART II - BUILD THE BUSINESS CASE

- 04 - Dimension 1: Cost Structure
- 05 - Dimension 2: Scale Economics
- 06 - Dimension 3: The Value of Control
- 07 - Dimension 4: Dependency & Continuity
- 08 - Dimension 5: Strategic Flexibility
- 09 - Dimension 6: Responsibility Cost

### PART III - CALCULATE THE DECISION

- 10 - Build a 3-Year and 5-Year TCO Model
- 11 - Calculate Value, Not Only Cost
- 12 - Three Illustrative Business Scenarios
- 13 - When Self-Hosting Does Not Make Financial Sense
- 14 - Build the Executive Business Case
- 15 - Evaluate the Open VTS Business Case

## PART I

# The Economics Behind the Hosting Model

Before comparing vendors, separate the software quote from the operating system of costs around it. The goal is to understand what each hosting model makes visible, what it bundles, and what it transfers to your organization.

**PRICE / COST STRUCTURE / TCO**

These are the ideas to carry into the next section.

**01 - PRICE VS COST**

## GPS Software Cost Is More Than the Price Tag

The number printed on a proposal is important. It is also incomplete.

A platform can create cost before deployment, during migration, throughout daily operations, and when the organization eventually changes direction. Some of those costs appear on a vendor invoice. Others appear in cloud bills, staff time, integration projects, storage growth, support contracts, or operational friction.

### A useful cost review separates seven layers:

- Software or subscription
- Infrastructure and storage
- Implementation and migration
- Operations and administration
- Support and updates
- Integrations, data, and third-party services
- Change, replacement, and exit

The purpose is not to make every hidden cost sound negative. Bundled costs can be valuable when they transfer responsibility to a provider. Direct costs can be valuable when they purchase control the business genuinely uses.

**Core idea**

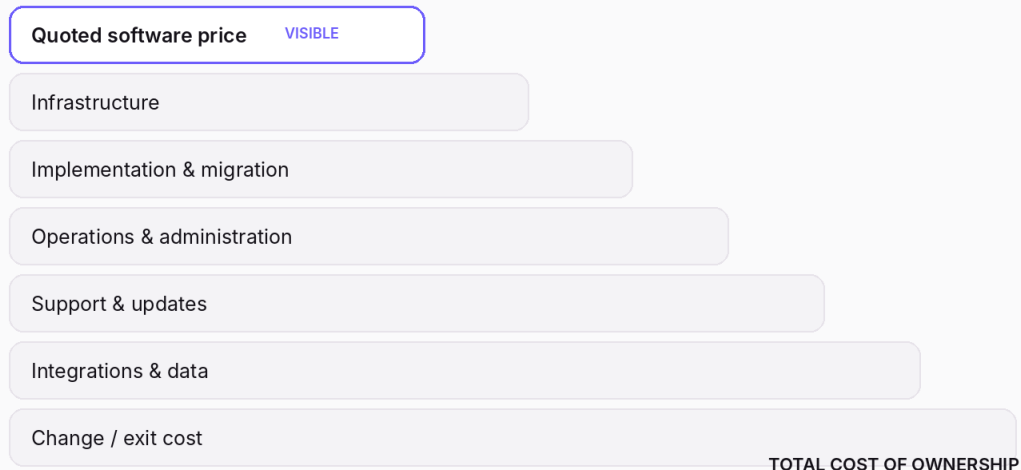
Price is what appears on the quote. Total cost is what the operating model requires over time.

## 01 - PRICE VS COST

# What the Quote Does Not Show

## PRICE IS ONE LINE. TOTAL COST IS THE OPERATING MODEL.

A useful business case separates the visible quote from the costs required to operate the platform over time.



A good business case makes the full stack visible before asking which model is cheaper.

## 02 - OPERATING MODELS

# SaaS and Self-Hosted Create Different Cost Structures

Provider-operated software and self-hosted software allocate cost differently because they allocate responsibility differently.

In a SaaS model, application infrastructure, core operations, platform maintenance, and some resilience or security work are commonly bundled into the provider's recurring price. The customer pays for the outcome without directly operating most of the environment.

In a self-hosted model, the software and the infrastructure become more visible as separate economic components. The organization may pay a license or support fee while also purchasing compute, storage, backups, monitoring, networking, and administration.

## This can change the shape of cost in three ways:

- Fixed vs variable - some self-hosted costs may remain stable across a range of fleet sizes, while usage-linked fees grow directly with scale.
- Bundled vs explicit - SaaS can hide infrastructure complexity by bundling it; self-hosting makes more operational costs visible.
- Provider labor vs internal labor - activities performed by the SaaS provider may need to be performed, automated, or outsourced by the self-hosted customer.

The correct comparison must normalize these differences rather than compare a subscription fee with a license fee in isolation.

## 02 - OPERATING MODELS

# Compare the Structure, Not the Labels

## TWO OPERATING MODELS. DIFFERENT COST STRUCTURES.

Neither model is automatically cheaper. The business case depends on scale, requirements, and operating capability.

### PROVIDER-OPERATED / SaaS

#### Recurring platform fee

Commonly primary

#### Infrastructure

Bundled into service

#### Scaling

Often linked to usage / vehicles

#### Updates

Typically provider-managed

#### Internal operations

Lower direct infrastructure burden

### SELF-HOSTED

#### Software license

Separate from infrastructure

#### Infrastructure

Customer-selected and operated

#### Scaling

Capacity planned around workload

#### Updates

Customer deploys within support model

#### Internal operations

Higher direct responsibility

### Decision rule

Treat "managed" as a service you are buying and "control" as a responsibility you are accepting. Price both.

## 03 - TOTAL COST OF OWNERSHIP

# Build a Comparable TCO Model

Total Cost of Ownership (TCO) is useful because it forces the decision beyond the initial purchase. For this guide, TCO means the cost required to implement, operate, support, grow, and eventually change the tracking platform over a defined period.

**A practical GPS-platform TCO can be organized into eight categories:**

| Cost Category  | Examples                                         |
|----------------|--------------------------------------------------|
| Software       | License, subscription, plan, vehicle fees        |
| Infrastructure | Compute, storage, database, bandwidth, backup    |
| Implementation | Installation, configuration, migration, training |

| Cost Category     | Examples                                        |
|-------------------|-------------------------------------------------|
| Operations        | Administration, monitoring, incident handling   |
| Support & Updates | Vendor support, maintenance, upgrade work       |
| Integrations      | APIs, development, third-party services         |
| Data              | Retention, archiving, analytics, storage growth |
| Change / Exit     | Reconfiguration, export, migration, replacement |

A TCO model should have a clear time horizon - usually three or five years - and should state the assumptions that change the result most strongly: vehicle growth, reporting interval, data retention, infrastructure profile, labor cost, and vendor pricing.

### 03 - TOTAL COST OF OWNERSHIP

## The TCO Formula

Use a model simple enough to understand but complete enough to expose the operating differences.

**TCO =**

Software + Infrastructure + Implementation + Operations + Support + Integrations + Data + Change

### Then calculate it twice:

#### Current / provider-operated model

Use the actual recurring fees, included services, paid add-ons, internal administration, integration costs, and likely growth charges.

#### Self-hosted model

Use the actual license/support terms plus infrastructure, backups, monitoring, staff time, implementation, migration, storage, security, and upgrade assumptions.

### Avoid three common mistakes:

- Treating internal staff time as free
- Assuming infrastructure cost remains constant while telemetry and retention grow
- Using the first-year software price as if it were the five-year total cost

#### Keep strategic value separate

TCO compares cost. It does not fully capture control, dependency, faster integration, or business options. Those belong in the value side of the business case.

## PART II

# Build the Business Case

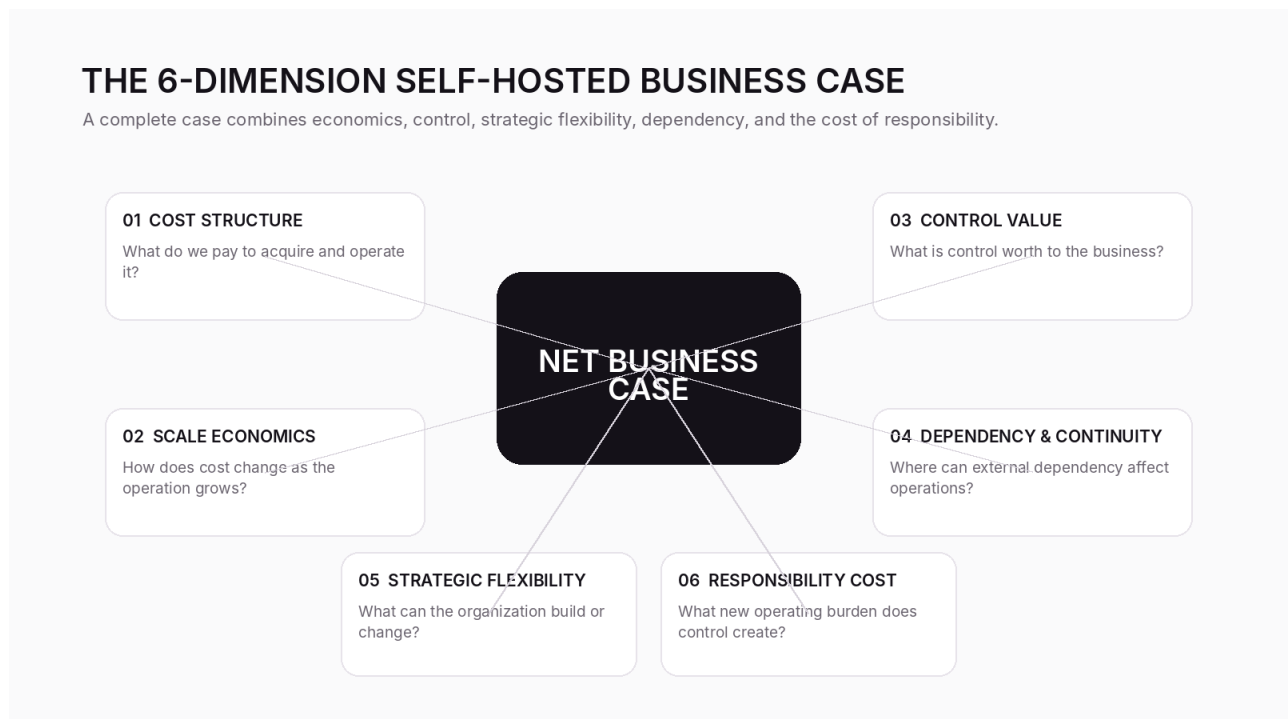
The financial model becomes useful only when it is combined with the value the operating model creates and the responsibility it transfers. These six dimensions form the decision framework.

## ECONOMICS / CONTROL / VALUE / RESPONSIBILITY

These are the ideas to carry into the next section.

## THE FRAMEWORK

## The 6-Dimension Self-Hosted Business Case



The six dimensions are designed to prevent two opposite errors: treating self-hosting as free control, or treating managed convenience as free of long-term dependency.

## DIMENSION 01

## Cost Structure

Start by mapping every cost to the party that pays it and the behavior that makes it change.

## For each cost item, record four properties:

| Question            | What to Record                                               |
|---------------------|--------------------------------------------------------------|
| Who pays it?        | Vendor, customer, cloud provider, internal team, third party |
| How does it behave? | Fixed, variable, stepped, one-time, usage-linked             |
| What drives it?     | Vehicles, telemetry, storage, users, support, integrations   |
| How certain is it?  | Contracted, measured, estimated, unknown                     |

## Typical SaaS cost drivers

- Per-vehicle or per-device recurring fees
- Higher tiers for features, users, data retention, or API usage
- Add-on charges for support, messaging, reports, storage, or integrations

## Typical self-hosted cost drivers

- Software license and update/support entitlement
- Compute, storage, backup, networking, monitoring
- Implementation, system administration, maintenance, and security

### Do not compare unlike numbers

A \$X monthly SaaS fee and a \$Y self-hosted license are not comparable until infrastructure, labor, support, and the time horizon are normalized.

### DIMENSION 01

# Build the Cost Inventory

Use this inventory before any ROI calculation. Replace every placeholder with a current quote, invoice, cloud estimate, or internal labor assumption.

| Cost Item          | Current Model       | Self-Hosted Model      | Evidence              |
|--------------------|---------------------|------------------------|-----------------------|
| Software / license | Recurring fee       | License / support      | Quote / terms         |
| Compute            | Bundled or indirect | Customer-selected      | Cloud/server estimate |
| Storage & backup   | Included / tiered   | Customer-managed       | Retention estimate    |
| Monitoring         | Bundled             | Internal / third-party | Tooling estimate      |
| Administration     | Low / moderate      | Direct responsibility  | Hours x labor rate    |
| Migration          | Future exit cost    | Initial implementation | Project estimate      |

| Cost Item    | Current Model           | Self-Hosted Model   | Evidence             |
|--------------|-------------------------|---------------------|----------------------|
| Integrations | Plan / vendor dependent | API / custom build  | Engineering estimate |
| Support      | Included / paid tier    | Support entitlement | Commercial terms     |

Once the inventory is complete, highlight the three variables most likely to change. Sensitivity analysis on those variables is often more valuable than adding dozens of minor line items.

## DIMENSION 02

# Scale Economics

The economic case can change dramatically as a tracking operation grows.

A recurring model linked directly to vehicles or usage may start with low implementation friction and predictable pricing. A self-hosted model may require greater upfront effort but distribute some costs across a larger operating base.

The important question is not whether one curve is always lower. It is whether your expected growth creates a meaningful crossover - and whether the organization is prepared to operate the model on the other side of that crossover.

## Model at least four scale points

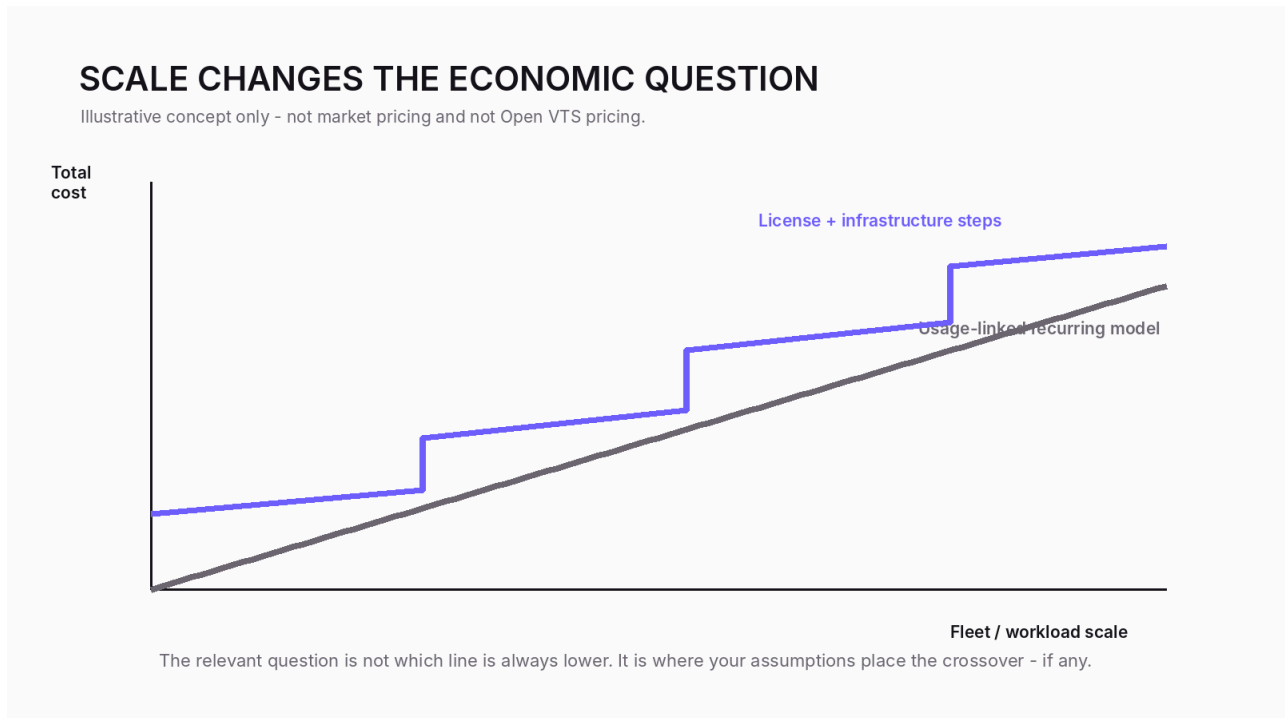
| Scale Point     | Questions to Recalculate                                                       |
|-----------------|--------------------------------------------------------------------------------|
| 100 vehicles    | Is self-hosting operationally justified at all?                                |
| 500 vehicles    | Do recurring platform fees begin to matter materially?                         |
| 2,500 vehicles  | Do infrastructure and administration scale slower than usage-linked fees?      |
| 10,000 vehicles | Do data, integrations, support, and reliability become strategic cost drivers? |

### Important

Vehicle count is not the only scale variable. Reporting interval, data retention, sensor payloads, API traffic, reports, users, and automation can change infrastructure cost substantially.

## DIMENSION 02

# Model the Crossover - If There Is One



A responsible business case allows for the possibility that no crossover exists within the organization's planning horizon.

## DIMENSION 03

## The Value of Control

Not every benefit of self-hosting reduces a software invoice. Some benefits create options elsewhere in the organization.

Control can have business value when it enables the organization to make decisions that would otherwise require provider approval, a higher service tier, custom vendor work, or a platform change.

### Areas where control may create value include:

- Infrastructure policy - choosing environment, capacity, network, and deployment location
- Data environment - shaping retention, backup, analytics, and surrounding access
- Integration architecture - connecting tracking data to internal business systems
- Upgrade timing - aligning maintenance with operational windows
- Operational design - building role structures, workflows, reports, or automation around the business
- Product strategy - using the platform as part of a GPS or telematics service offered to customers

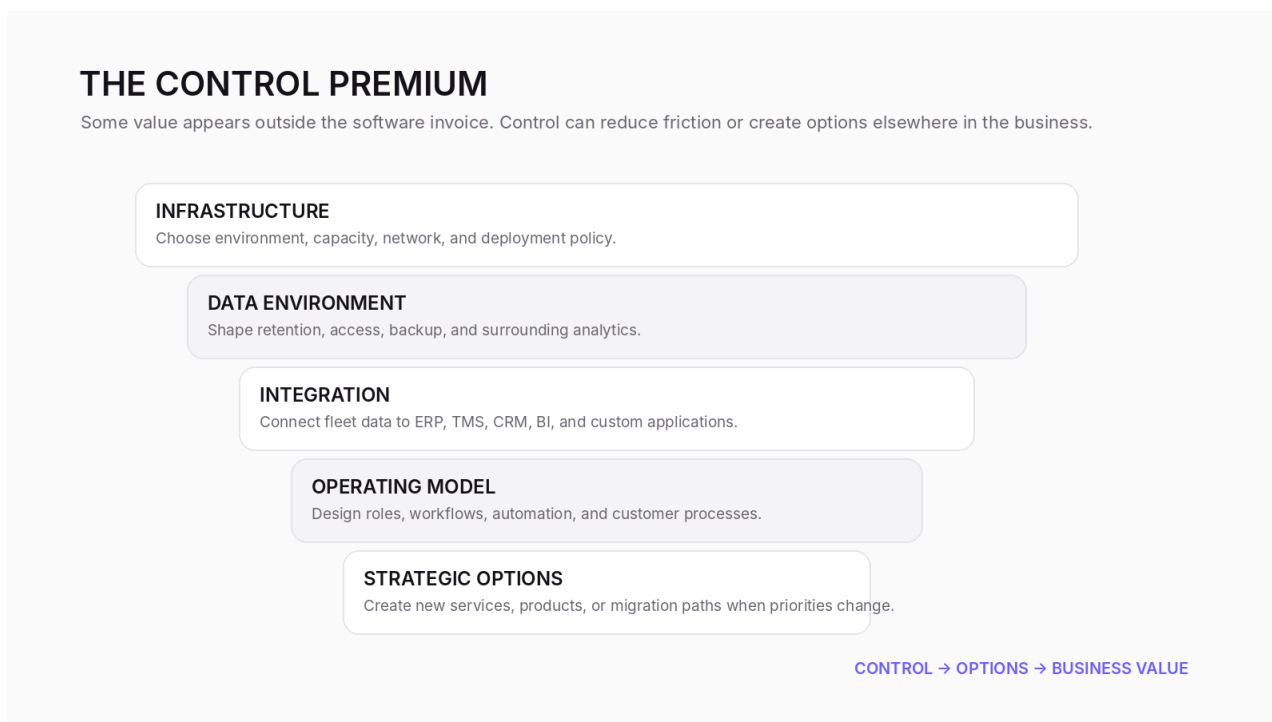
The business case should not automatically assign a dollar figure to these benefits. First determine whether the control will actually be used.

#### Control test

If we gained this control tomorrow, what would we do differently - and what measurable or strategic consequence would that create?

## DIMENSION 03

# The Control Premium



If the organization cannot identify a real use for control, do not treat control itself as financial value.

## DIMENSION 04

## Dependency & Continuity

Every vehicle tracking platform depends on software, infrastructure, people, networks, devices, and third-party services. The business question is not whether dependency exists. It is whether the important dependencies are understood and acceptable.

Provider-operated software may place more operational continuity inside the vendor's environment. Self-hosted software can move more of that continuity into infrastructure the customer controls, while software licensing, updates, support, maps, messaging, email, cloud services, or other dependencies may remain.

### Map six categories

- Software and licensing
- Infrastructure and hosting
- GPS devices and protocol availability
- Third-party services
- Integrations and data contracts
- People and operational skills

### For each dependency, record:

- What happens if it becomes temporarily unavailable?
- How quickly would operations be affected?

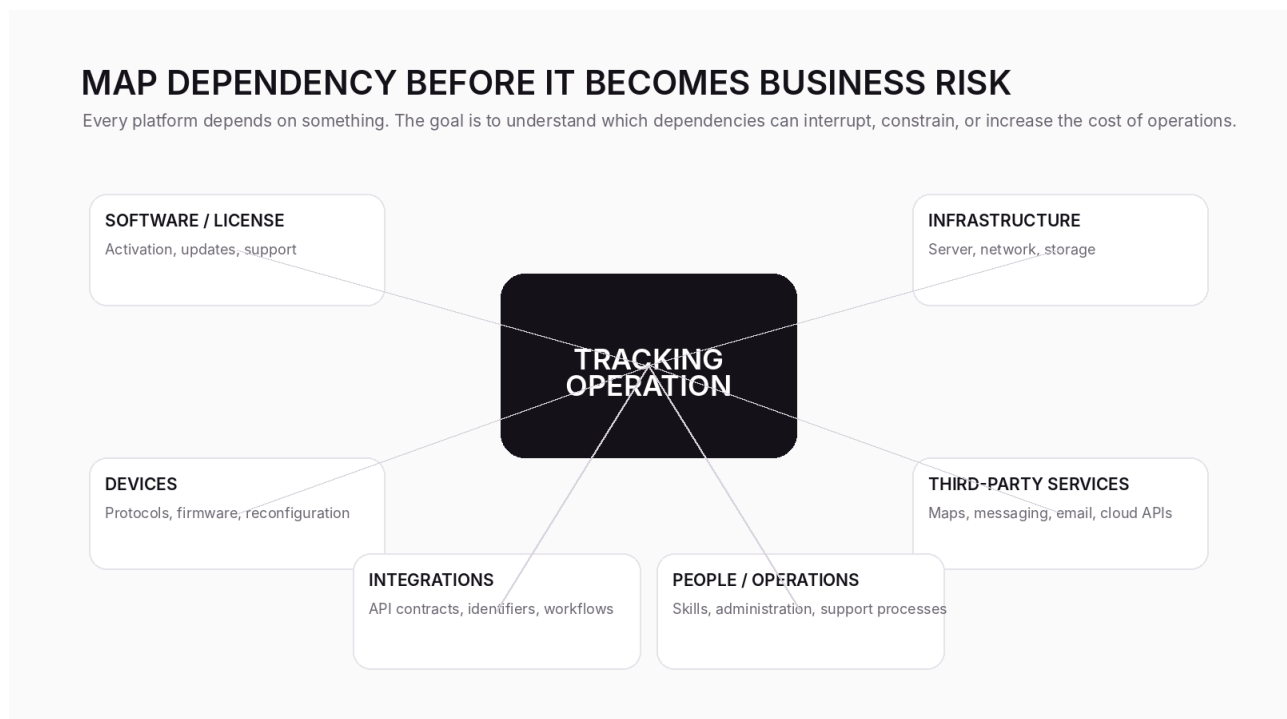
- Can the dependency be substituted?
- What is the recovery path?
- Who controls the contract or infrastructure?

### Continuity is not zero dependency

A self-hosted platform can increase operational control without becoming independent of every vendor or external service.

#### DIMENSION 04

## Turn Dependency Into a Map



Dependencies are easiest to manage when they are visible before a failure, price change, acquisition, or platform migration makes them urgent.

#### DIMENSION 05

## Strategic Flexibility

Strategic flexibility is the value of being able to change the tracking operation without replacing the entire platform.

For a GPS or telematics business, that can matter because the platform may sit underneath customer services, device programs, support processes, billing, mobile applications, or integrations. For an enterprise fleet, it may matter because tracking data increasingly participates in ERP, TMS, dispatch, BI, safety, or customer systems.

## Evaluate flexibility across five areas:

| Area           | Business Question                                                                   |
|----------------|-------------------------------------------------------------------------------------|
| Devices        | Can new tracker protocols or hardware families be added when requirements change?   |
| Integrations   | Can fleet data participate in other business systems through documented interfaces? |
| Automation     | Can repeatable operational rules be turned into workflows?                          |
| Applications   | Can customer or internal applications use the platform data and services?           |
| Infrastructure | Can capacity and surrounding services evolve with the operation?                    |

Strategic flexibility is not automatically ROI. It becomes economic value when it reduces future project cost, accelerates a launch, avoids a forced platform migration, or enables revenue or efficiency that would otherwise be difficult.

### DIMENSION 05

## From Platform Control to Business Options

Open VTS provides useful examples of how platform architecture can create strategic options, without requiring the buyer to value those options automatically.

### Examples to verify in an Open VTS evaluation include:

- Protocol-driven device ingestion for a broad GPS-device ecosystem
- A process for implementing and testing an additional documented device protocol when required
- HTTP APIs and realtime interfaces for external applications
- Workflow automation that can evaluate conditions and execute actions
- Customer-controlled infrastructure that can be sized around the expected workload
- A separately open-source mobile application that can be used as part of the mobile strategy

### The correct business question is:

**Which of these capabilities will materially reduce cost, create revenue, improve service, or reduce future dependency in our organization?**

Only those capabilities should be counted as strategic value in the final business case.

## DIMENSION 06

# Responsibility Cost

This is the chapter that prevents the business case from becoming self-hosting propaganda.

When an organization controls more of the environment, someone must operate that environment. The responsibility may be performed by internal IT, a managed infrastructure partner, automation, or a combination of all three. It still has a cost.

## Price these responsibilities explicitly:

- Server provisioning and operating-system maintenance
- Firewall, DNS, TLS, and administrative access
- Database and storage administration
- Backup and restore testing
- Monitoring, health checks, and capacity alerts
- Application updates and maintenance windows
- Security operations and credential management
- Incident response and escalation
- Internal technical support and vendor coordination

A common mistake is to include cloud compute but exclude the people required to operate it. Another is to assume that one person will spend "almost no time" on administration without measuring the real workload.

### Responsibility principle

Control is not free. The business case is strongest when the value of control exceeds the cost of operating it.

## DIMENSION 06

# Price the Responsibility

## CONTROL HAS AN OPERATING COST

The self-hosted business case is incomplete until these responsibilities have owners, time estimates, and cost assumptions.

### SERVER & OS

Provisioning, patching, capacity

### NETWORK & TLS

Firewall, DNS, certificates, access

### BACKUP & RECOVERY

Backups, restore tests, retention

### MONITORING

Health, queues, storage, alerts

### APPLICATION MAINTENANCE

Updates, change windows, validation

### PEOPLE & SUPPORT

Admin time, expertise, incident handling

[ASSIGN OWNER](#) · [ESTIMATE TIME](#) · [PRICE THE RESPONSIBILITY](#)

For every box, assign an owner, estimate monthly effort, and identify the evidence behind the estimate.

## PART III

# Calculate the Decision

The six dimensions now become a model. Use a multi-year TCO, a separate strategic-value assessment, realistic scenarios, and explicit risk to create an executive decision that can survive scrutiny.

## 3-YEAR TCO / 5-YEAR TCO / VALUE / RISK

These are the ideas to carry into the next section.

## 10 - MULTI-YEAR TCO

# Build a 3-Year and 5-Year Model

One year is often too short to evaluate a platform whose implementation, data growth, integrations, and operational dependence accumulate over time.

## Build at least two horizons:

### 3-year model

Useful for near-term budget planning, implementation payback, and realistic organizational forecasts.

### 5-year model

Useful for understanding scale, storage growth, major updates, infrastructure refresh, and the long-term effect of recurring pricing.

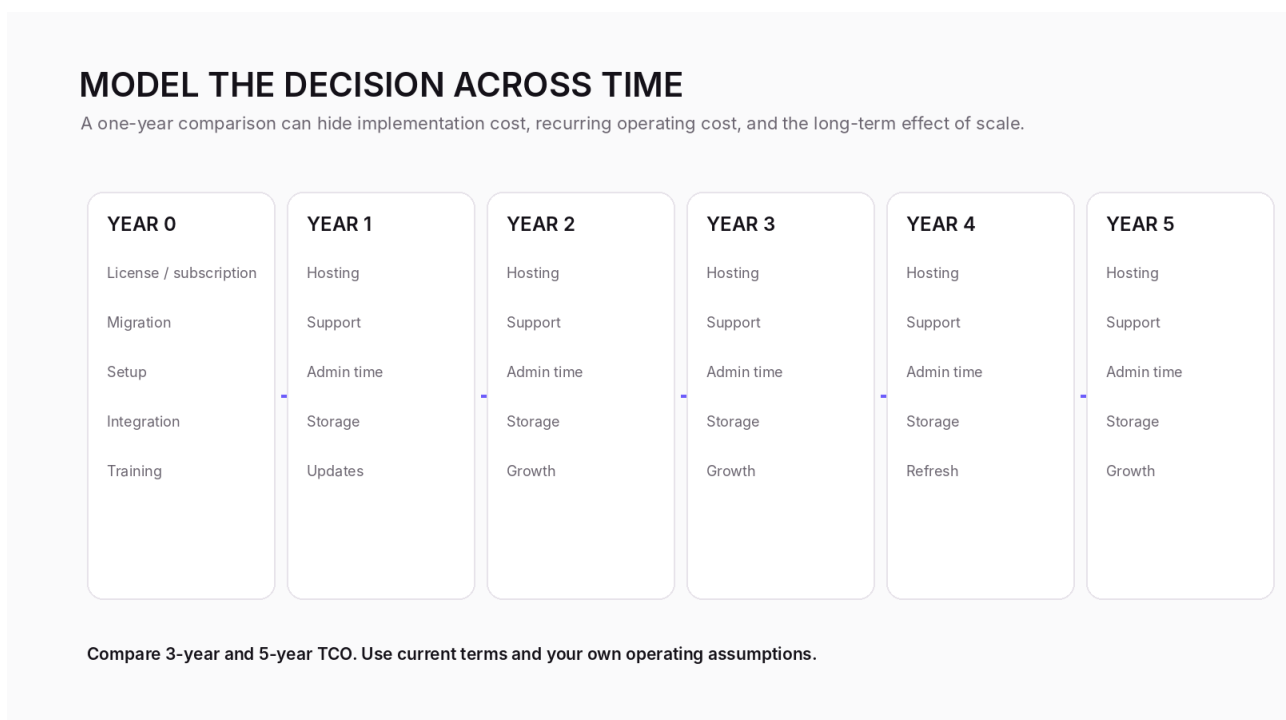
## Separate the model into:

- Year 0 implementation costs
- Annual recurring operating costs
- Variable costs that grow with vehicles, usage, or storage
- Step costs such as server upgrades or additional administration
- One-time change costs such as migration or major integration work

Use a base case, conservative case, and high-growth case. Leadership should be able to see which assumptions change the result most strongly.

### 10 - MULTI-YEAR TCO

## Model the Timeline



The final calculator should show both cumulative cost and annual cost so the organization can see when a crossover occurs and what creates it.

### 11 - VALUE

## Calculate Value, Not Only Cost

A platform decision can create value even when the software itself is not cheaper.

The discipline is to separate measurable value from strategic value.

## Measurable value

- Avoided or reduced recurring platform charges
- Reduced manual administration through automation
- Reduced integration work because APIs expose required data
- Avoided hardware replacement when existing protocols can be supported
- Reduced duplicate data entry or reconciliation
- Reduced third-party service cost where the operating model changes those services

## Strategic value

- Infrastructure control
- Data-environment control
- Freedom to design integration architecture
- Ability to build customer-facing services
- Greater control over upgrade timing
- Reduced exposure to a particular hosted operating model

If a strategic benefit cannot be quantified honestly, score it rather than monetize it. A transparent 4/5 strategic score is more credible than an invented \$100,000 benefit.

### 11 - VALUE

## Use a Strategic Value Score

One practical approach is to maintain financial ROI and strategic value as two separate outputs.

| Strategic Dimension       | Weight | Score 1-5 | Evidence / Reason |
|---------------------------|--------|-----------|-------------------|
| Infrastructure control    | 15%    |           |                   |
| Data environment          | 20%    |           |                   |
| Device flexibility        | 15%    |           |                   |
| Integration freedom       | 20%    |           |                   |
| Automation / operations   | 15%    |           |                   |
| Continuity / exit options | 15%    |           |                   |

## Then present leadership with two conclusions:

### Financial case

3-year and 5-year TCO, annual cash requirement, and break-even if applicable.

**Strategic case**

Weighted value of control, flexibility, continuity, and operating fit.

The decision is strongest when both point in the same direction. When they conflict, leadership can see the trade-off explicitly instead of hiding it inside one ROI percentage.

**12 - SCENARIO A**

## Small Internal Fleet - 100 Vehicles

Illustrative profile only - not market pricing and not Open VTS pricing.

A business operates approximately 100 vehicles. Tracking requirements are straightforward: live location, basic history, alerts, and standard reporting. There are few integrations, limited custom workflows, and no plan to sell tracking services to customers.

### Likely economic characteristics

- Recurring SaaS cost may remain manageable because the fleet is small
- Infrastructure administration may represent a meaningful share of total self-hosted cost
- The organization may gain little measurable value from deep platform control
- Internal technical capability may be more expensive than the software savings

### Questions that matter

- Is there a compliance, data, integration, or infrastructure requirement that materially benefits from self-hosting?
- Can the organization administer the environment efficiently?
- Would a managed service allow the operations team to focus on higher-value work?

**Possible conclusion**

Self-hosting may still be strategically justified, but a cost-only case may be weak at this scale if the organization has simple requirements and no existing infrastructure capability.

**12 - SCENARIO B**

## Growing GPS Operator - 2,000 Vehicles

Illustrative profile only - not market pricing and not Open VTS pricing.

A GPS service business manages approximately 2,000 vehicles across multiple customers and several tracker families. It operates support, users, reports, customer access, and integrations. The fleet is expected to grow materially over the next three years.

### Likely economic characteristics

- Vehicle-linked recurring software fees may become a material operating expense
- Some infrastructure and administration costs can be distributed across a larger vehicle base
- Device compatibility and protocol flexibility can reduce hardware-change costs
- APIs, team permissions, automation, and branding may have direct commercial value
- Platform dependency becomes part of the business model rather than merely an IT concern

## Questions that matter

- At what fleet size does the multi-year TCO comparison change?
- How much administration is genuinely required?
- Which integrations or workflows would create measurable labor savings?
- What is the value of controlling the platform offered to customers?

### Possible conclusion

This is the type of profile where the economics and strategic value of self-hosting often deserve detailed modeling rather than a simple monthly-price comparison.

## 12 - SCENARIO C

# Large Telematics Operation - 10,000+ Vehicles

Illustrative profile only - not market pricing and not Open VTS pricing.

A telematics operation manages more than 10,000 vehicles with frequent telemetry, multiple customer organizations, APIs, automation, mobile users, long data retention, and several third-party systems.

## Likely economic characteristics

- Software pricing is only one component of a much larger operating architecture
- Data volume, retention, database design, and infrastructure capacity become major cost drivers
- Integration and automation can create measurable efficiency across teams
- Reliability, observability, backup, and incident response require professional operations
- Vendor dependency and exit planning become strategic governance concerns

## Questions that matter

- Can the organization operate the infrastructure more efficiently than the bundled managed-service premium?
- Which costs grow with telemetry and storage rather than vehicle count?
- What additional staff or managed infrastructure support is required?
- What business risk is reduced by controlling more of the operating environment?

### Possible conclusion

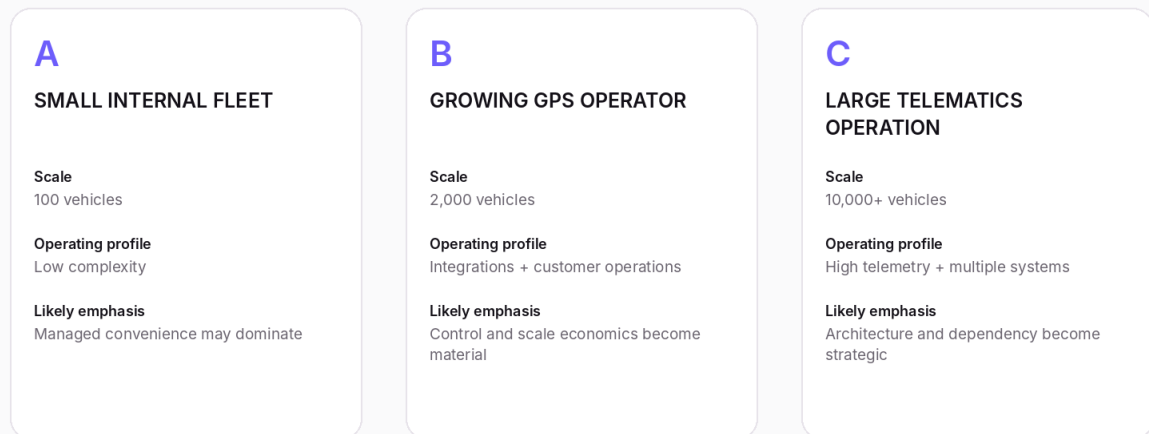
At large scale, the business case is no longer simply SaaS fee versus server cost. It becomes an architecture, control, operations, and continuity decision with financial consequences.

## 12 - SCENARIOS

# Scale Changes What Matters

## THE SAME HOSTING MODEL CAN PRODUCE DIFFERENT BUSINESS CASES

Illustrative profiles - not pricing claims or recommendations.



The scenario that resembles your organization should determine which assumptions receive the most scrutiny.

## 13 - DECISION DISCIPLINE

# When Self-Hosting Does Not Make Financial Sense

A credible business case must include reasons not to proceed.

## Self-hosting may be economically unattractive when:

- The fleet is small and recurring platform cost is already low
- The organization has no technical operating capability and must purchase expensive managed administration
- There is little need for custom integration, automation, data control, or device flexibility
- Infrastructure/security requirements create more cost than the control is worth
- The expected planning horizon is short
- Migration effort is high relative to the remaining value of the current platform
- A provider-operated service bundles support, resilience, infrastructure, and compliance capabilities more efficiently than the organization can reproduce

Self-hosting may also be strategically unnecessary even when it is financially possible.

### Important

A business case is more trustworthy when "do not proceed" is a valid outcome.

## 13 - DECISION DISCIPLINE

## Use Go / Investigate / No-Go Signals

| Signal      | Meaning                                                                                                                                     |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| GO          | Multi-year economics are favorable, strategic value is material, responsibilities have owners, and the PoC validates technical assumptions. |
| INVESTIGATE | The model is sensitive to a few uncertain variables, technical ownership is incomplete, or migration/integration risk needs proof.          |
| NO-GO       | Mandatory requirements fail, operating responsibility is not viable, the financial case is weak, or risk exceeds the value of control.      |

**Before recommending self-hosting, leadership should be able to answer:**

- Who owns infrastructure operations?
- What is the 3-year and 5-year cost?
- Which strategic benefits are actually relevant?
- What happens under high-growth assumptions?
- Which dependencies remain?
- What evidence supports the workload and staffing assumptions?
- What is the migration and rollback plan?

**If one of these answers is still "we assume," the business case is not finished.**

## 14 - EXECUTIVE CASE

## Build the Management Recommendation

Executives usually do not need forty pages of platform detail. They need a decision that shows the economics, strategic rationale, new responsibility, major risk, and evidence.

**A strong executive business case can fit on one page using six blocks:**

| Block            | What It Should Answer                                                    |
|------------------|--------------------------------------------------------------------------|
| Current State    | What do we operate today, what does it cost, and what constraints exist? |
| Proposed State   | What operating model are we considering and why?                         |
| Financial Effect | 3-year TCO, 5-year TCO, break-even, annual cash requirement              |

| Block                 | What It Should Answer                                              |
|-----------------------|--------------------------------------------------------------------|
| Strategic Effect      | Control, flexibility, integration, continuity, business options    |
| New Responsibilities  | Infrastructure, security, backup, monitoring, updates, support     |
| Recommendation & Risk | Proceed, PoC, defer, or reject - with the largest unresolved risks |

The recommendation should be short enough that a finance leader can understand it without learning the entire GPS architecture, while retaining links to the evidence behind every material assumption.

#### 14 - EXECUTIVE CASE

## The Business Case Canvas

### CURRENT STATE

Current fleet size, current provider/model, recurring cost, existing infrastructure, major constraints.

### TARGET STATE

Proposed self-hosted architecture, expected scale, required device ecosystem, integrations, and operating model.

### ECONOMICS

3-year TCO / 5-year TCO / annual cost / break-even / sensitivity variables.

### STRATEGIC VALUE

Control, integration, device flexibility, data environment, continuity, product options.

### NEW RESPONSIBILITY

Server, network, security, backup, monitoring, updates, administration, support.

### DECISION

GO / INVESTIGATE / NO-GO + top three risks + next proof required.

## 15 - OPEN VTS

# Apply the Same Business Case to Open VTS

Open VTS should not be treated as the automatic winner of a framework published by Open VTS. Use the same assumptions, cost inventory, responsibility model, and evidence requirements applied to any alternative.

## Areas to include in an Open VTS business case

| Dimension               | Open VTS Evaluation Area                                                                        |
|-------------------------|-------------------------------------------------------------------------------------------------|
| Cost Structure          | Current license, support/update entitlement, deployment infrastructure, internal administration |
| Scale Economics         | Server capacity, telemetry workload, storage, growth, and current commercial terms              |
| Control Value           | Customer-controlled deployment environment and operational data services                        |
| Dependency & Continuity | Licensing, infrastructure, third-party services, support, maps/messaging where used             |
| Strategic Flexibility   | Device protocol ecosystem, APIs, realtime interfaces, workflows, mobile application strategy    |
| Responsibility Cost     | Server, OS, firewall, backup, monitoring, updates, security, incident handling                  |

The core Open VTS platform is proprietary and self-hosted. The mobile application is separately open source. Do not translate "self-hosted" into unsupported claims about ownership of the core application source code.

## 15 - OPEN VTS

# What to Verify Before Approving the Case

Use current evidence rather than this eBook for changing product or commercial details.

## Verify the following:

- Current Open VTS pricing and license terms
- Current update and support entitlement
- Target Windows, Linux, or Docker deployment requirements
- Server sizing for the expected telemetry workload
- Backup, retention, monitoring, and recovery responsibilities
- Exact GPS-device protocol compatibility and required commands/sensors
- API operations and realtime interfaces required by integrations
- Workflow capabilities required by the operating model
- Open-source mobile application terms and implementation requirements
- Any third-party services used for maps, email, messaging, push notification, or other functions

Open VTS evidence should come from the released product, current documentation, API reference, supported-device resources, current commercial terms, and a proof of concept using the buyer's actual requirements.

### Do not use a static ROI claim

Open VTS should provide the inputs and evidence. The buyer should calculate the ROI with their own fleet, costs, infrastructure, staffing, and growth assumptions.

## ACTION CHECKLIST

# Before You Present the Business Case

- ☐ Document current recurring and hidden operating costs.
- ☐ Choose a 3-year and 5-year planning horizon.
- ☐ Forecast fleet growth and telemetry workload.
- ☐ Price the self-hosted infrastructure realistically.
- ☐ Include staff or managed-administration cost.
- ☐ Include backup, monitoring, security, and update work.
- ☐ Include migration and integration cost.
- ☐ Separate measurable savings from strategic value.
- ☐ Map licensing, infrastructure, device, and third-party dependencies.
- ☐ Identify the top three assumptions that can change the result.
- ☐ Run conservative, base, and high-growth scenarios.
- ☐ Assign an owner to each operating responsibility.
- ☐ Run a proof of concept for technical assumptions.
- ☐ Review current commercial and legal terms.
- ☐ Prepare a one-page executive canvas.
- ☐ Make GO, INVESTIGATE, and NO-GO valid outcomes.

**A business case is ready when leadership can see both what the organization gains and what it must operate in return.**

## GLOSSARY & REFERENCES

# Terms and Sources

## Key Terms

**Break-Even** - The point at which cumulative value or savings equals the additional cost of an alternative.

**Fixed Cost** - A cost that remains broadly stable across a defined operating range.

**Variable Cost** - A cost that changes with vehicles, usage, data, users, or another activity driver.

**TCO** - Total Cost of Ownership - the cost to implement, operate, support, grow, and change a system over a defined period.

**ROI** - Return on Investment - a financial comparison between value generated and investment required.

**Self-Hosted** - Software deployed within infrastructure selected or controlled by the customer.

**SaaS** - Software as a Service - software primarily operated by the provider and consumed as a hosted service.

**Strategic Value** - A benefit important to the organization that may not be responsibly convertible into a direct financial amount.

**Telemetry** - Operational data transmitted by tracking devices, such as location, speed, ignition, sensors, or events.

## Further Reading

FinOps Foundation - FinOps Framework: <https://www.finops.org/framework/>

AWS Well-Architected Framework - Cost Optimization Pillar: <https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/welcome.html>

Microsoft Azure Well-Architected Framework - Cost Optimization: <https://learn.microsoft.com/azure/well-architected/cost-optimization/>

Google Cloud Architecture Framework - Cost Optimization: <https://cloud.google.com/architecture/framework/cost-optimization>

NIST Cybersecurity Framework 2.0: <https://www.nist.gov/cyberframework>

Open VTS - Product and resources: <https://openvts.io>

Open VTS - API Reference: <https://openvts.io/resources/api-reference>

Open VTS - Supported Devices: <https://openvts.io/product/supported-devices>

Open VTS - Pricing: <https://openvts.io/pricing>

Open VTS - Security: <https://openvts.io/security>



OPEN VTS / GUIDE 02 OF 06

# Your Infrastructure. Your Fleet Data. Your Rules.

Open VTS is a self-hosted vehicle tracking platform designed for GPS and telematics businesses, fleet operators, integrators, and organizations that want greater control over their tracking infrastructure and operational data environment.

The platform combines device-protocol connectivity, live fleet operations, reporting, organizational access control, APIs, realtime communication, workflow automation, and web/mobile experiences within a customer-controlled deployment model.

## OPEN VTS

Track Without Limits. Powered By You. | <https://openvts.io> | 2026 Edition | Version 1.1