

Sample

PRACTICAL HANDBOOK

Exit-Ready SaaS Infrastructure

István Pató

A practical handbook for SaaS founders
and buyers building transferable systems.

TwoMinuteCode Series

Table of Contents

Exit-Ready SaaS Infrastructure	2
1 Introduction	4
1.1 Considerations	6
1.1.1 Time Is Money	6
1.1.2 Customer Security and Brand Reputation	6
1.1.3 Trust Matters	6
1.2 Topics Covered	7
1.3 Topics Not Covered	8
1.4 Before You Begin	10
2 Ownership Fundamentals	12
2.1 Account Strategy	13
2.1.1 What to Do at the Beginning	13
2.1.2 Access and Recovery	14
2.1.3 Documentation	14
3 Identity and Communication	16
3.1 Google Account and Gmail	17
3.1.1 Security and Recovery	17
3.1.2 Using Your Own Domain	19
3.1.3 Handover	19
3.2 Google Chrome	22
3.2.1 Make It More Convenient	23
3.2.2 Handover	23

4 Source Code and Package Management	24
4.1 GitHub	25
4.1.1 Create an Organization	25
4.1.2 Handover	26
4.2 npmjs.com	29
4.2.1 Signing In	29
4.2.2 Publishing	30
4.2.3 Handover	31
5 Domain and Brand Ownership	32
5.1 Namecheap	33
5.1.1 Handover	34
6 Social Media and Content	35
6.1 YouTube	36
6.1.1 Handover	36
6.2 X	37
6.2.1 Handover	37
6.3 LinkedIn Page	39
6.3.1 Handover	39
6.4 TikTok	41
6.4.1 Handover	41
6.5 Facebook	43
6.5.1 Handover	43
7 Application Infrastructure	45
7.1 Vercel	46
7.1.1 Handover	46
7.2 Supabase	48
7.2.1 Handover	48
7.2.1.1 Supabase Organization Handover	48
7.2.1.2 Project Handover	49

7.3 Bunny.net	51
7.3.1 Handover	51
8 Email Infrastructure	53
8.1 Resend	54
8.1.1 Handover	54
9 Finance and Sales	56
9.1 Revolut	57
9.2 Polar	58
9.2.1 Handover	59
10 Analytics and Observability	60
10.1 PostHog	61
10.1.1 Handover	61
Closing Remarks	64
About the Author	65



Exit-Ready SaaS Infrastructure

This book explains how to build SaaS infrastructure that does not depend solely on its founder and can be transferred to a buyer when necessary, such as during a sale. It is written for both buyers and sellers, offering practical advice and guidance for a smooth handover.

For corrections, outdated steps, or unclear sections, email feedback@transferreadysaas.com. Feedback is reviewed for future updates, but this address does not include individual SaaS consulting or technical support.

Legal Notice

Copyright © 2026 István Pató. All rights reserved.

No part of this publication may be reproduced, stored, or transmitted without the author's prior written permission, except for brief quotations and uses permitted by law.

This book is intended for educational and informational purposes. It does not constitute legal, financial, tax, or information security advice. You are responsible for evaluating the described solutions in your own environment and consulting an appropriate professional when necessary.

The product and company names, logos, and trademarks mentioned in this book are the property of their respective owners. Their inclusion does not imply endorsement or an official relationship.

PART 1



Introduction

Sample

The value of a SaaS business is not determined solely by its source code and revenue. Ownership of its domains, service accounts, social media profiles, and application infrastructure is equally important.

1.1

Considerations

1.1.1 Time Is Money

Design your infrastructure carefully from the beginning so that you do not have to waste effort restructuring it later. Such changes may require a substantial migration or cause a service disruption.

1.1.2 Customer Security and Brand Reputation

Imagine that you have an established brand and a successful SaaS service, only for a competitor to claim your brand name on X. That is not even the worst-case scenario. What if a malicious actor registers the npmjs.com scope that matches your brand name and publishes dangerous code that your customers download and use, perhaps as part of your SaaS solution? It would be a nightmare.

1.1.3 Trust Matters

By planning for transferability in advance, you significantly increase the buyer's confidence. As the previous example shows, this is also important for security and may become a deciding factor for the buyer.

1.2

Topics Covered

I will show you how to configure the services required by your SaaS system so that they remain easy to manage and secure as your company grows, while also being straightforward to transfer if you later sell the system.

You will also learn why you should reserve your brand name on certain services even if you do not currently use them.

A SaaS system may rely on many services, from domain registration and email delivery to databases. I cannot cover every system, but this book describes the services I use and those that are widely adopted.

I wrote this book to share my experience so that, regardless of which systems you use to build your SaaS, you will consider transferability when making decisions and configuring services.

I use the following services for my SaaS application: **Gmail, GitHub, npmjs.com, YouTube, X, LinkedIn, TikTok, Facebook, Namecheap, Revolut, Vercel, Supabase, Resend, Polar, PostHog, Bunny.net**

The list alone can be useful if you are currently planning a SaaS infrastructure because it provides a reliable starting point.

1.3

Topics Not Covered

The scope of this book is intentionally focused. It does not provide detailed setup guides or explain how to secure and configure every individual service. When you need that information, consult the service's documentation or my upcoming books.

Even so, this book remains useful because service documentation rarely addresses transferability from this perspective.

What does it mean to be exit-ready?

The following criteria apply to the **services used by your SaaS system**. They do not cover the transferability of the SaaS product itself, which has many additional requirements.

- ownership is clearly defined;
- the brand name is reserved on security-critical services;
- you control the account associated with the brand name on major social media platforms;
- the infrastructure does not depend on personal accounts;
- access is properly protected;
- the services are well documented and regularly reviewed;
- the services can be transferred to a new owner with minimal disruption.

The list may sound like a long and exhausting set of tasks, but it is not. You need to perform the same tasks when building your SaaS anyway. The only difference is that **you must do them properly**.

The goal is to build a safer business that can operate independently

of its founder and to ensure that the SaaS can be sold later.

1.4

Before You Begin

The book covers two main topics:

1. creating and configuring services so that they can be transferred;
2. the handover itself, including practical steps and advice.

Create a plan and schedule for the handover.

- document exactly what will be transferred, when, by whom, and to whom, and have both parties approve it;
- work together with the buyer during the handover, which may take several days;
- notify users of service disruptions and define maintenance windows for the transitions;
- account for delays caused by parties other than the buyer and seller, and agree on how to handle revenue earned during the transition;
- prepare for a transition period during which the buyer already owns the infrastructure components but you still need to provide technical assistance;
- remember that many services, such as Vercel and Supabase, can be connected to a GitHub repository. Map the development-time dependencies between infrastructure components. This is not the software architecture; **it is the architecture of the development environment;**
- consider GDPR and other legal requirements during the handover, and proceed carefully;

Do not expect everything to succeed in a single day. It is better to pro-

ceed carefully than to make mistakes. The descriptions in this book may not always match your exact needs because every starting situation is different. For example, whether the buyer acquires your company or only the SaaS product can affect changes to billing and payment settings.

Review the resources at <https://help.acquire.com> as well. You will find useful advice there whether you are a buyer or a seller.