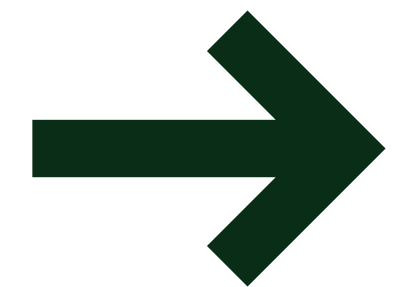


True Open Source, True Freedom: The Frappe Framework Story

Your paragraph text

Exploring the principles of software freedom and how the Frappe Framework empowers communities through true open source solutions.



Ramakrishnan Nataraj

CEO, KlyONIX Tech Consulting Pvt Ltd

The Reality Check

What Developers Face Today?

01

"Open Source": Many platforms claim open source but lock critical features behind paid walls

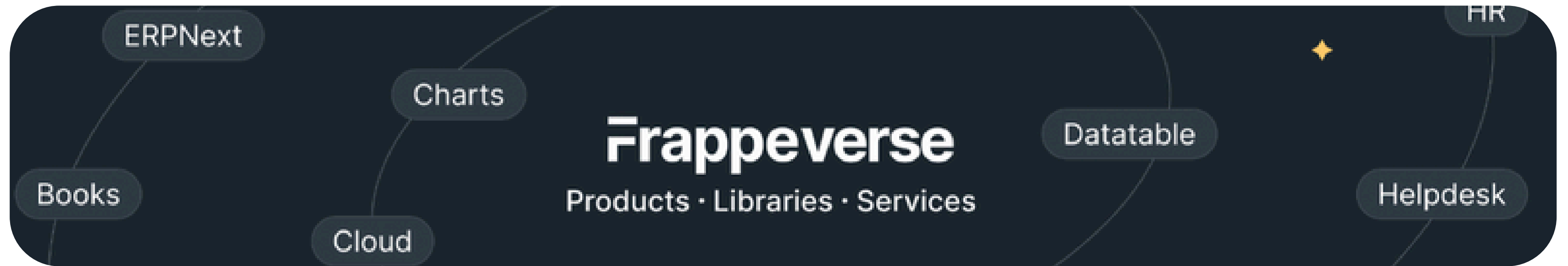
02

Licensing Confusion: Dual licensing that changes rules when you succeed

03

Developer Frustration: Spending more time fighting frameworks than building solutions

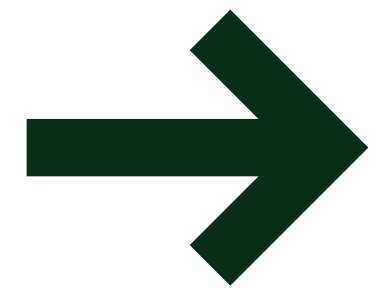
The Promise: "Build fast with low-code!" The Reality: "Pay more for basic features"



Frappe Framework

A full stack framework, batteries-included, and written in Python and JS.

Frappe, pronounced fra-pay, is a full stack, batteries-included, web framework written in Python and Javascript with MariaDB as the database. It is the framework which powers ERPNext, is pretty generic and can be used to build database driven apps.



Building Custom Applications

Extending Functionality

Customizing UI and Forms

Creating Reports and Dashboards

Developing APIs

Managing Databases

Implementing Security

Frappe True Open Source by Design



MIT License Framework

Core framework remains truly free

Full Code Access

Every line of code is inspectable
and modifiable



No Hidden Costs

All enterprise features included out-of-the-box

"Enterprise technology can be open, accessible, and sustainable"

The Developer's Dilemma Solved

Low-Code + Full-Code = True Flexibility

Traditional Problem:

Low-Code Platform: "Build fast but can't customize"

Full-Code Framework: "Full control but slow development"

Frappe Solution:

```
# Start with low-code (DocType)
frappe.get_doc({
    'doctype': 'Custom Task',
    'title': 'Build MVP',
    'status': 'In Progress'
}).insert()

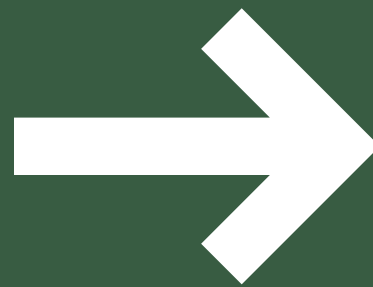
# Scale with full-code when needed
class CustomTask(Document):
    def validate(self):
        # Your custom business logic here
        if self.priority == "Critical":
            self.send_urgent_notification()
```

DocTypes - The Swiss Army Knife

One Concept, Infinite Possibilities

What is a DocType?

- **Model:** Database table definition
- **View:** Auto-generated forms and lists
- **Controller:** Python class for business logic
- **API:** REST endpoints created automatically



Define "Project" DocType → Get:

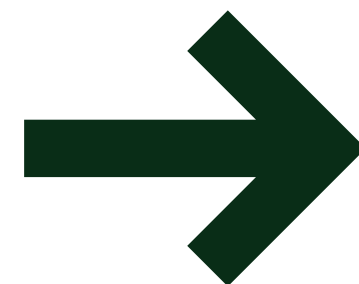
- ✓ Database table
- ✓ Web forms
- ✓ Mobile-ready UI
- ✓ REST API endpoints
- ✓ Search functionality
- ✓ Permission system

Metadata-Driven Development

Just Configure, Don't Code (Until You Need To)

Traditional Way:

```
-- Create table
CREATE TABLE projects (...);
-- Build forms with HTML/CSS
-- Write API endpoints
-- Handle permissions
-- 100+ lines of code
```



Frappe Way:

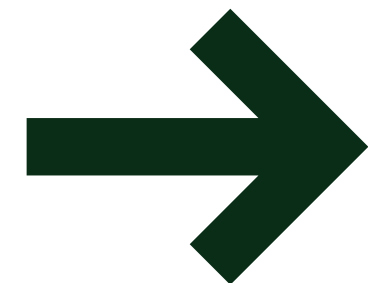
```
// Just metadata
{
  "doctype": "Project",
  "fields": [
    {"fieldname": "title", "fieldtype":
>Data"},
    {"fieldname": "status",
'fieldtype': "Select"}
  ]
}
// Everything else is auto-generated!
```




Enterprise-Grade Security

Built for Mission-Critical Applications

- Role-Based Access Control: Field-level permissions
- Data Encryption: At rest and in transit
- Audit Trails: Every change tracked
- Session Management: Secure authentication
- IP Whitelisting: Network-level security
- Two-Factor Authentication: Built-in 2FA support



The Ecosystem Advantage

Core Applications:

ERPNext: Complete ERP solution

Frappe CRM: Customer relationship management

Frappe HR: Human resource management

Frappe LMS: Learning management system

Gameplan: Work communication tool

Community Benefits:

Active Forums: 50,000+ developers

Documentation: Comprehensive guides

Training Programs: Certification available

Partner Network: Global implementation support

The Future is Open

What's Coming Next

Enhanced Frappe UI : Modern, responsive design system
Mobile-First: Progressive web app improvements





Get Involved Today!

 Try Now: frappe.io/framework

 Learn: docs.frappe.io

 Community: discuss.frappe.io

 Deploy: frappe.cloud

