A Swiss Cloud-Native Farm 🐄

Clément Nussbaumer

SWISS
**CLOUD
NATIVE
DAY**

clement.n8r.ch

About Me

👨‍💻 Clément Nussbaumer

- Platform Engineer at PostFinance 🏦
- 6+ years with Kubernetes ☸️
- Golang enthusiast 🐹
- Musician in a Brass Band 🎺

🚜 My Wife's Family Farm

- Dairy operation in rural Switzerland 🇨🇭
- Biogas plant ⚡
- 65 cows 🥛
- 250 chicken 🥚
- Work (silage, etc.) for third parties 🌾

Recent selfie with Jura (cow #33) 🐄

The Challenge

The Challenge:

- Legacy systems with old protocols 📟
- Outdated/slow GUIs 🐌
- Limited monitoring 📊

Goal:

- Real-time data collection ⚡
- Customized alerting 🔔

"Swiss Farm" (1883) by Eugène Burnand

Oil on canvas, farm in Ecublens near Seppey, Vaud

The Farm

Main Site

65 cows, dairy facilities, milk vending machine

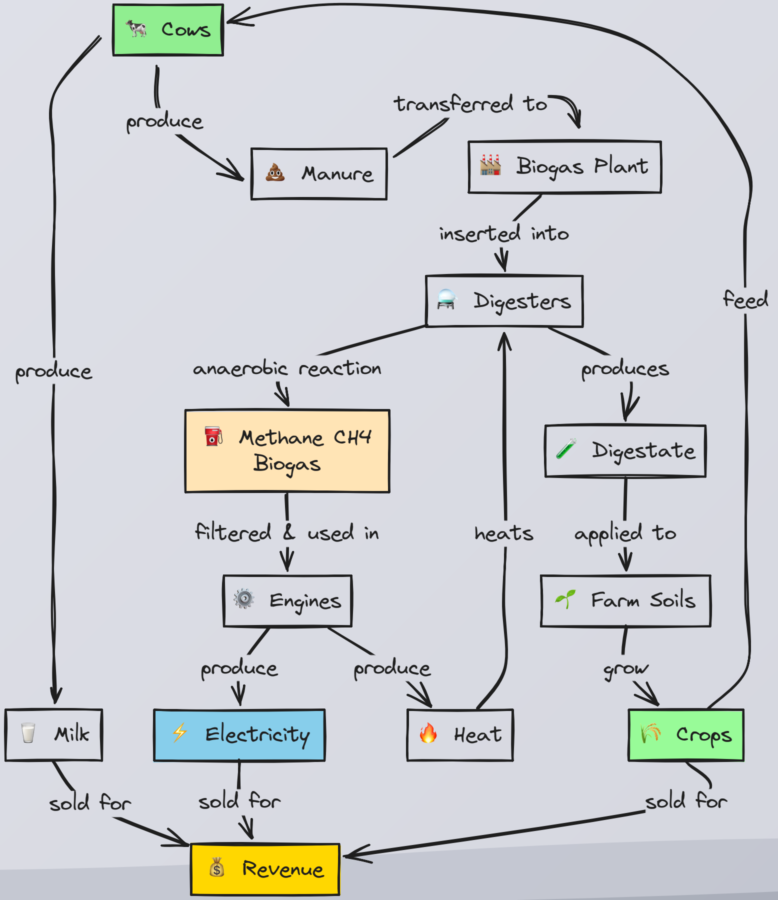
Biogas Plant Site

Energy production

The Biogas Cycle

A complete **circular economy** on the farm:

- Waste becomes energy
- Residual heat is used to heat up the digesters
- … and to dry up crops
- Solid digestate is used as fertilizer to further grow crops



Cows

produce

Manure

transferred to

Biogas Plant

inserted into

Digesters

produce

anaerobic reaction

produces

feed

Methane CH4
Biogas

Digestate

filtered & used in

heats

applied to

Engines

Farm Soils

produce

produce

grow

Milk

Electricity

Heat

Crops

sold for

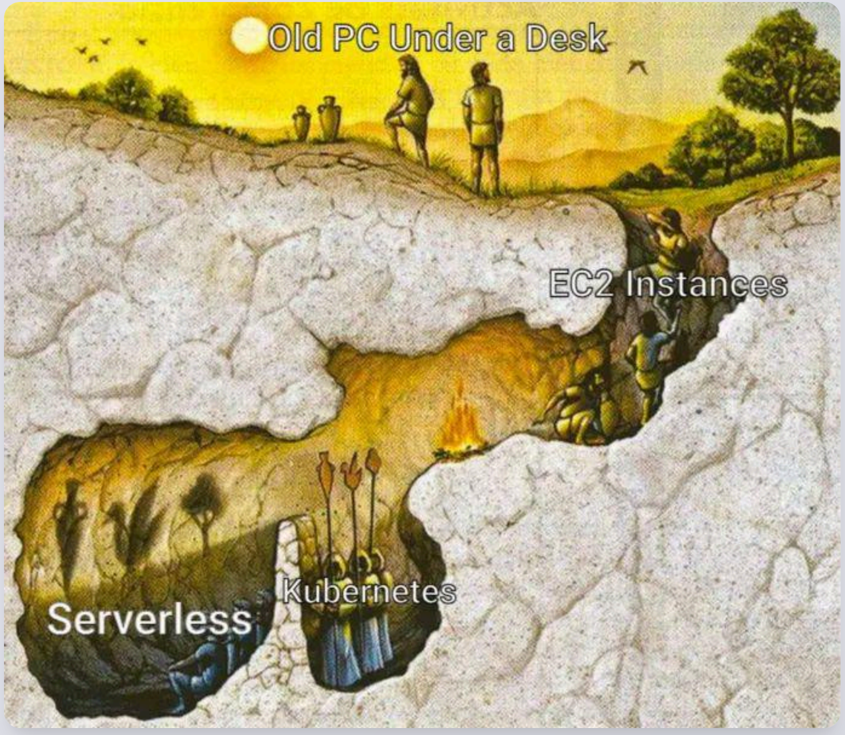
sold for

sold for

Revenue

Let's Turn the Farm Cloud Native

Infrastructure Foundation



Kubernetes Cluster

- 4x HP EliteDesk Mini PCs
- Intel i5/i7 processors
- 16GB RAM each
- 256GB-512GB NVMe storage
- Talos Linux OS

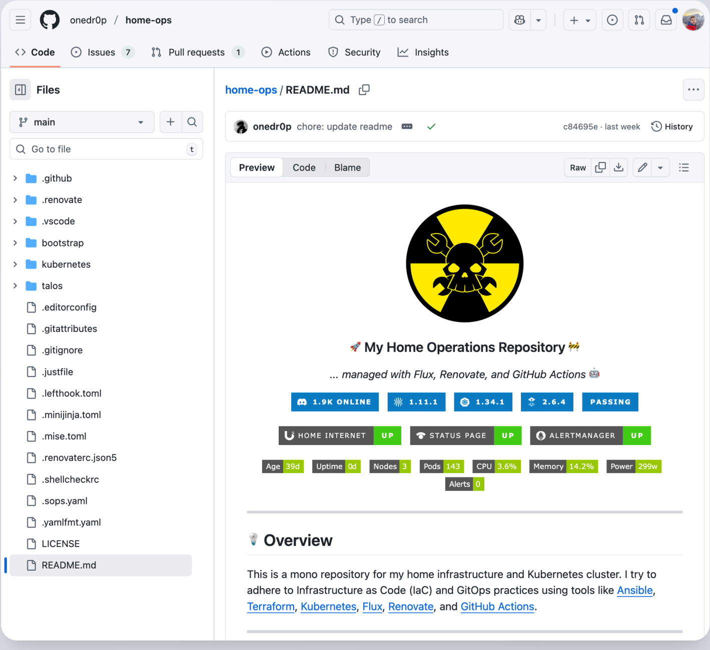
Operating Costs

- Power: 1.4 kWh/day @ 26.87¢ = **137 CHF/year**
- Hardware amortization: **250 CHF/year** (=1 node)
- **Total: 387 CHF/year (~32 CHF/month)**

Home Operations Community

Inspired by Home Operations[1]

- "We love to break production at home" 🏠
- Key contributors: `bjw-s`, `onedr0p`
- **GitOps**-first approach with FluxCD
- Encrypted secrets with SOPS
- Open-source home lab configurations
- Active Discord & GitHub community



.github
.renovate
.vscode
bootstrap
kubernetes
talos
.editorconfig
.gitattributes
.gitignore
.justfile
.lefthook.toml
.minijinja.toml
.mise.toml
.renovaterc.json5
.shellcheckrc
.sops.yaml
.yamlfmt.yaml
LICENSE
README.md

home-ops / README.md

onedr0p chore: update readme ✓ c84695e · last week

🚀 My Home Operations Repository 🚧

... managed with Flux, Renovate, and GitHub Actions 🤖

1.9K ONLINE | 1.11.1 | 1.34.1 | 2.6.4 | PASSING

HOME INTERNET UP | STATUS PAGE UP | ALERTMANAGER UP

Age 39d | Uptime 0d | Nodes 3 | Pods 143 | CPU 3.6% | Memory 14.2% | Power 299w | Alerts 0

💡 Overview

This is a mono repository for my home infrastructure and Kubernetes cluster. I try to adhere to Infrastructure as Code (IaC) and GitOps practices using tools like Ansible, Terraform, Kubernetes, Flux, Renovate, and GitHub Actions.

github.com/onedr0p/home-ops

[1] https://home-operations.com/

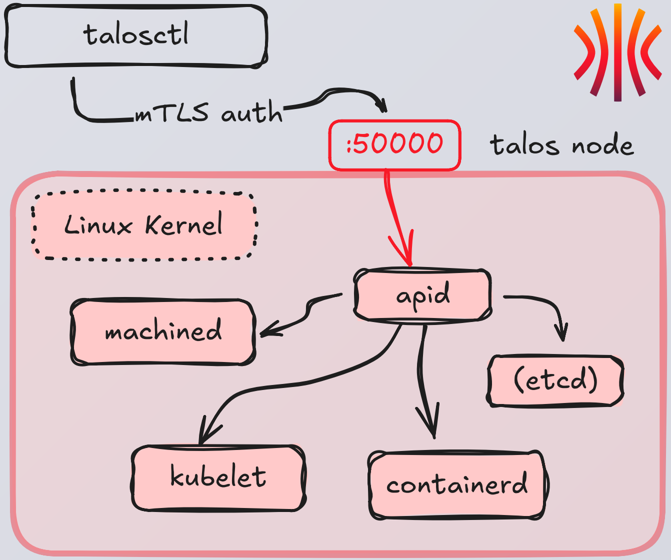
Talos Linux

The 12 binaries’ O.S.[1]

immutable, minimal, secure

declarative configuration file and gRPC API [2]

```
$ talosctl services
SERVICE                  STATE      HEALTH
apid                     Running    OK
containerd               Running    OK
cri                      Running    OK
etcd                     Running    OK
kubelet                  Running    OK
machined                 Running    OK
syslogd                  Running    OK
trustd                   Running    OK
udevd                    Running    OK
```



[1] https://www.siderolabs.com/blog/there-are-only-12-binaries-in-talos-linux/

[2] https://www.talos.dev/v1.11/learn-more/philosophy/

Network Architecture

Dual-Stack Cluster

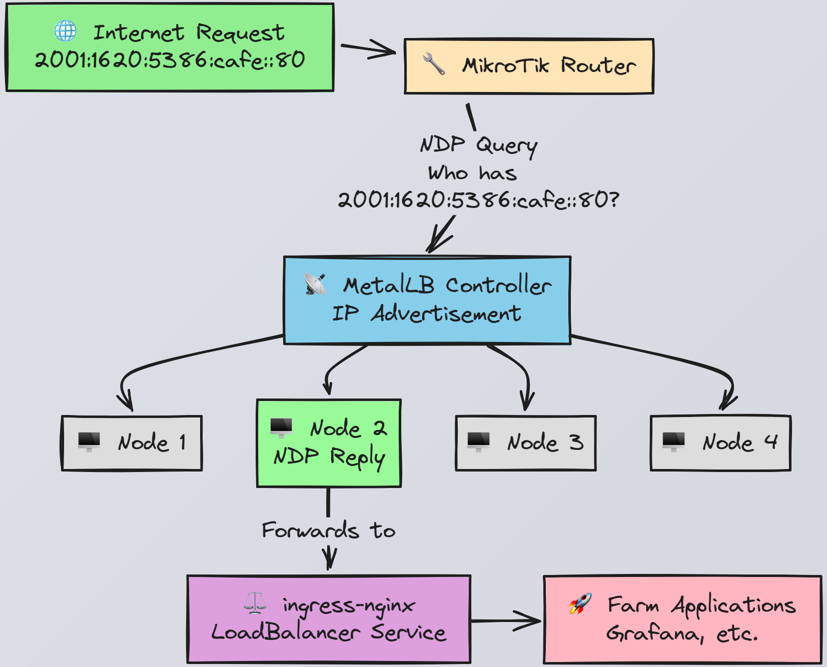
- Init7 🐇 copper connection 250/70MBps
- IPv6 `2001:1620:5386::/48` range:

  `2001:1620:5386:0000:0000:0000:0000:0000` -

  `2001:1620:5386:ffff:ffff:ffff:ffff:ffff`

MetalLB Load Balancer

- Routable IPv6 service range
- L2 Advertisement via NDP protocol
- Zero-cost HA with node failover
- no need for port-forwarding/NAT with IPv6



Stateful Storage

Distributed Storage

- Longhorn for replicated block storage
- VictoriaMetrics PVCs for time-series data
- HA MariaDB & PostgreSQL databases

Backup Strategy

- PVC snapshots via Velero and copied to offsite S3
- Database-specific backups[1]:

```
mariadb-dump | gzip --rsyncable && restic backup
```

- Incremental backups every 15 minutes

[1] https://clement.n8r.ch/en/articles/backing-up-mariadb-on-kubernetes/

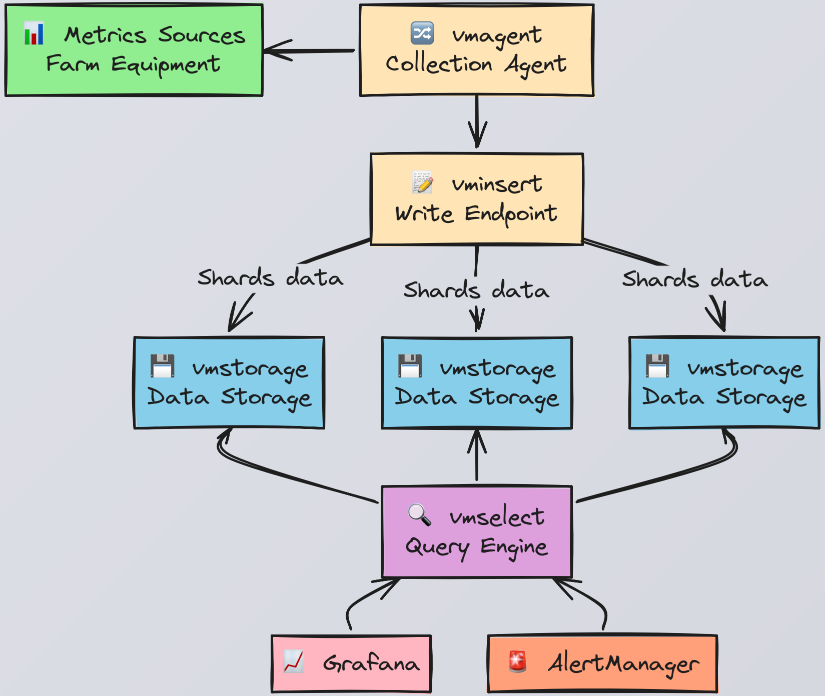
Monitoring Stack

Why VictoriaMetrics?

- Better resource usage than Prometheus
- Simple architecture
- Easy horizontal scaling
- Long-term data retention
- Drop-in Prometheus replacement

Grafana Integration

- Farm operations dashboards
- Custom alerting rules



Long-term Storage Strategy

Two VictoriaMetrics Clusters[1]

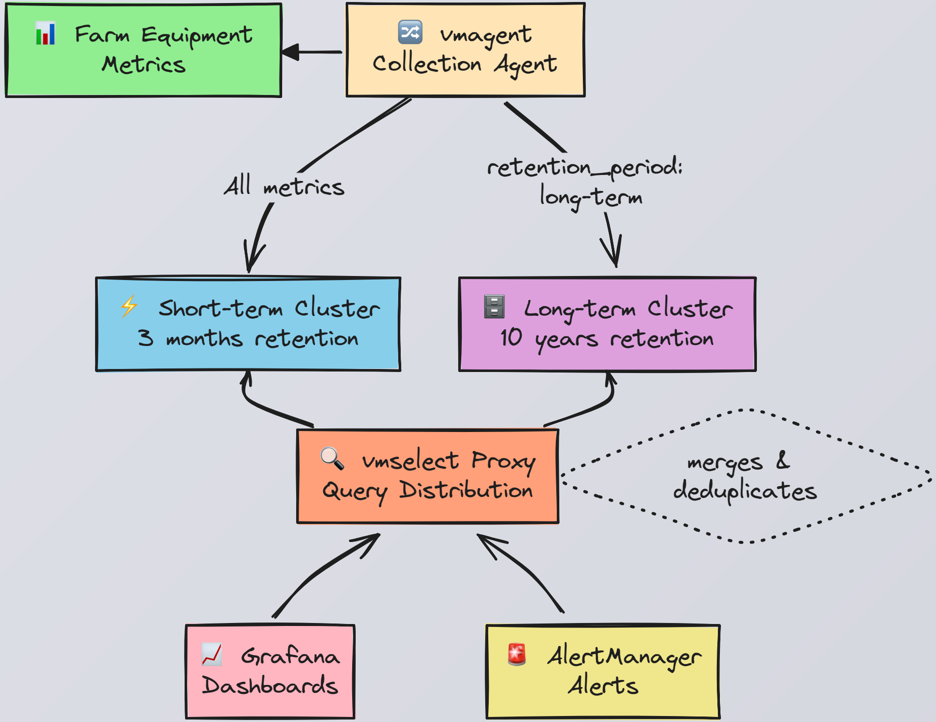
- **Short-term**: 3 months retention
- **Long-term**: 10 years retention

Smart Routing

- vmagent sends to long-term only if `retention_period=long-term`

vmselect Proxy

- Distributes and deduplicates queries across clusters


📊 Farm Equipment Metrics
🔀 vmagent Collection Agent
All metrics
retention_period: long-term
⚡ Short-term Cluster 3 months retention
🗄 Long-term Cluster 10 years retention
🔍 vmselect Proxy Query Distribution
merges & deduplicates
📈 Grafana Dashboards
🚨 AlertManager Alerts

[1] https://github.com/clementnuss/k8s-gitops/tree/main/workloads/metrics

Use Cases

🥛 Milking Data

Real-time collection & analysis

⚡ Biogas Plant

Performance metrics & alerts

🚧 Electric Fences

Remote control & monitoring

📊 Milk Vending Machine

LiDAR sensor integration

🧾 Invoicing Solution

Billing for e.g.silage work

DeLaval Milking Robot

DeLaval VMS™ Installed December 2024

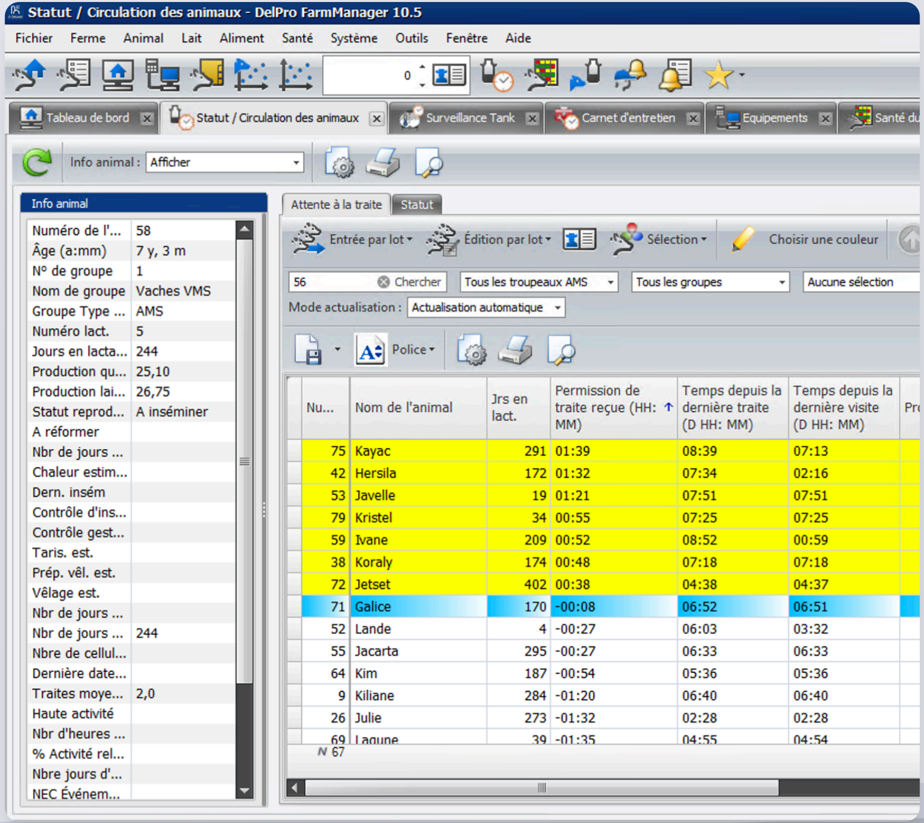
- Automated milking system
- 65 cows, 2.4 milkings per day average

DelPro Farm Manager Software

- Individual cow tracking & health monitoring
- Centralized herd management system
- Health & reproduction tracking
- Feeding optimization

Milking Robot

DelPro Farm Manager



Info animal : Afficher

Info animal	
Numéro de l'...	58
Âge (a:mm)	7 y, 3 m
N° de groupe	1
Nom de groupe	Vaches VMS
Groupe Type ...	AMS
Numéro lact.	5
Jours en lacta...	244
Production qu...	25,10
Production lai...	26,75
Statut reprod...	A inséminer
A réformer	
Nbr de jours ...	
Chaleur estim...	
Dern. insém	
Contrôle d'ins...	
Contrôle gest...	
Taris. est.	
Prép. vêl. est.	
Vêlage est.	
Nbr de jours ...	
Nbr de jours ...	244
Nbre de cellul...	
Dernière date...	
Traites moye...	2,0
Haute activité	
Nbr d'heures ...	
% Activité rel...	
Nbre jours d'...	
NEC Événem...	

56 Chercher | Tous les troupeaux AMS | Tous les groupes | Aucune sélection

Mode actualisation : Actualisation automatique

Nu...	Nom de l'animal	Jrs en lact.	Permission de traite reçue (HH: MM) ↑	Temps depuis la dernière traite (D HH: MM)	Temps depuis la dernière visite (D HH: MM)	Pr
75	Kayac	291	01:39	08:39	07:13	
42	Hersila	172	01:32	07:34	02:16	
53	Javelle	19	01:21	07:51	07:51	
79	Kristel	34	00:55	07:25	07:25	
59	Ivane	209	00:52	08:52	00:59	
38	Koraly	174	00:48	07:18	07:18	
72	Jetset	402	00:38	04:38	04:37	
71	Galice	170	-00:08	06:52	06:51	
52	Lande	4	-00:27	06:03	03:32	
55	Jacarta	295	-00:27	06:33	06:33	
64	Kim	187	-00:54	05:36	05:36	
9	Kiliane	284	-01:20	06:40	06:40	
26	Julie	273	-01:32	02:28	02:28	
69	Lagune	39	-01:35	04:55	04:54	

N 67

Legacy Management Interface

- Traditional Windows-based application
- Limited real-time capabilities
- Manual data export processes

Key Limitations

- No API access for automation
- Limited mobile access

DelPro Data Extraction

Custom-Built DelPro Exporter[1]

- Golang service that queries MS SQL database

Live Mode: `/metrics`

- Real-time Prometheus exposition format
- Current milking data and cow status

Historical Mode: `/historical-metrics`

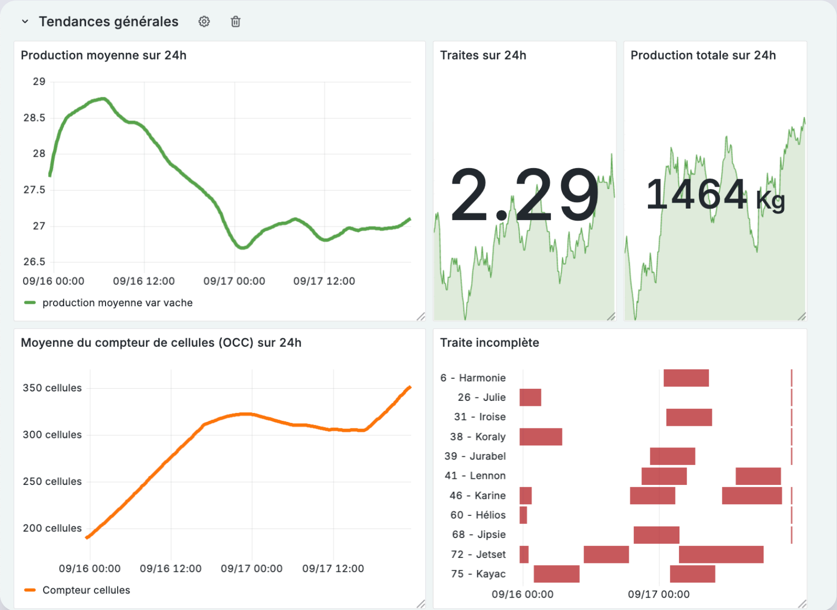
- Prometheus format with timestamps
- Backfilling historical data into VictoriaMetrics[2]

```go
// Query DelPro MS SQL database
records := db.Query(`
  SELECT animal_number, milk_yield, duration
  FROM milking_sessions
  WHERE session_end > ?
`, lastUpdate)

// Convert to Prometheus metrics
for _, r := range records {
  metrics.GetOrCreateGauge(
    "milk_yield_liters",
    map[string]string{
      "animal": r.AnimalNumber,
    }).Set(r.Yield)
}
```

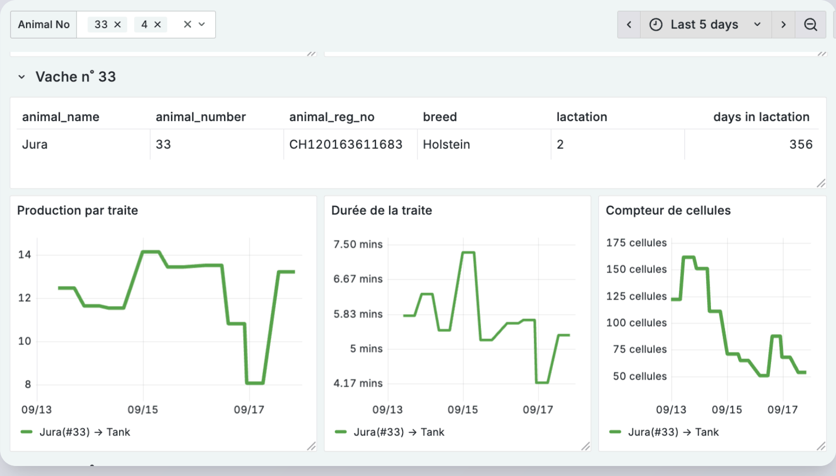
[1] https://github.com/clementnuss/delpro-exporter

[2] https://docs.victoriametrics.com/victoriametrics/#how-to-import-data-in-prometheus-exposition-format

Grafana Dashboard



Production trends and analytics



Animal No 33 × 4 ×

Last 5 days

animal_name	animal_number	animal_reg_no	breed	lactation	days in lactation
Jura	33	CH120163611683	Holstein	2	356

Individual cow stats - Jura (#33) 🐄

Biogas Plant Operations

External biogas facility

Control systems & monitoring

Engine overview

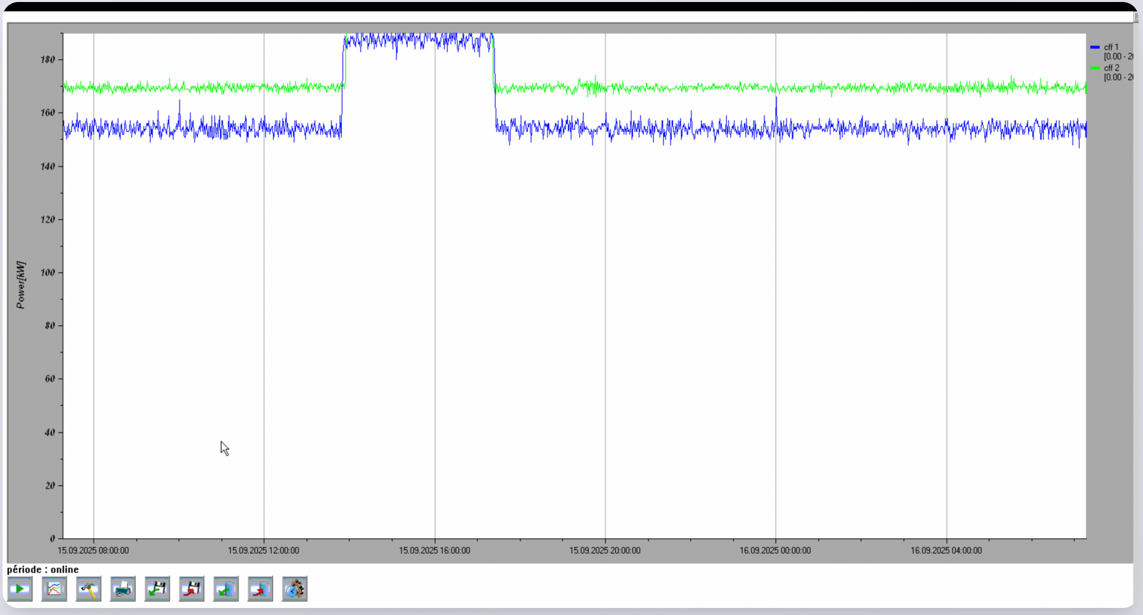
Energy Production

- 2x220 kW electrical capacity
- Waste-to-energy conversion
- Grid-connected power generation
- Heat recovery for farm operations

Key Metrics to Monitor

- Temperature profiles
- Electrical output
- Digester volume

SAIA PCD Controller GUI



Historical Data Visualization

Legacy Control Interface

- Windows-based management software
- Manual data export only

Key Limitations

- Limited remote capabilities (VNC only)
- Data locked in legacy system
- Outdated dashboard capabilities

SAIA EtherSBus Protocol

SAIA PCD Communication

- Proprietary EtherSBus protocol
- Network-based communication (UDP, port 5050)

Existing Python Library

- Found `digimat-saia` library
- Implements EtherSBus protocol

```python
# SAIA EtherSBus communication
from digimat_saia import SAIANode

# Create local node and declare remote server
node = SAIANode(253)
server = node.servers.declare('192.168.1.100')

# Read biogas parameters from remote PCD
gas_flow = server.registers[1000].float32
temperature = server.registers[1001].float32
power_output = server.registers[1002].float32
pump_running = server.flags[10].value
```

> **Problem: I don’t want to implement a prometheus exporter in Python again.**[1]

[1] https://github.com/clementnuss/alpro-openmetrics-exporter

gRPC Service Implementation

SAIA gRPC Service[1]

- Python service using digimat-saia
- ConnectRPC framework (CNCF sandbox)

ConnectRPC Benefits

- Simplified gRPC development

Service Capabilities

- Real-time parameter reading/setting
- Error handling & retries

```python
# gRPC service using ConnectRPC
class SaiaService(SaiaServiceServicer):
    def ReadFlag(
        self,
        request: saia_pb2.ReadFlagRequest,
        context: grpc.ServicerContext
    ) -> saia_pb2.ReadFlagResponse:
        r = typing.cast(
            SAIAItemFlag,
            server.flags[request.address]
        )

        if r is None or not r.isAlive():
            context.abort(
                grpc.StatusCode.INTERNAL,
                "unable to read register value"
            )

        return saia_pb2.ReadFlagResponse(value=r.bool)
```

[1] https://github.com/clementnuss/saia-grpc-service/blob/main/saia_grpc_service.py

Prometheus Metrics Exporter

SAIA PCD Exporter[1]

- Golang service for metrics collection
- Queries gRPC service periodically
- uses `VictoriaMetrics/metrics` go library

Data Pipeline Flow

1. SAIA PCD → EtherSBus protocol
2. gRPC service → JSON/Protobuf
3. delpro-exporter → metrics format
4. VictoriaMetrics → storage
5. Grafana → visualization

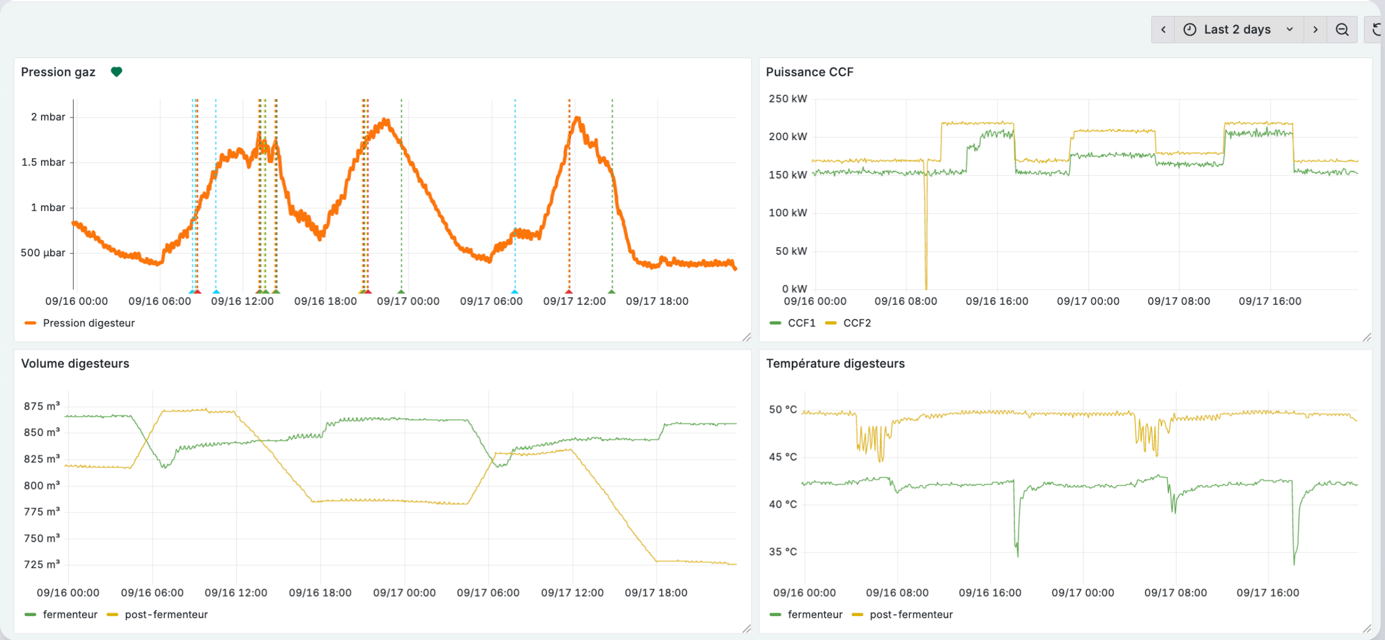
```go
// Query gRPC service and export metrics
func (e *Exporter) collectMetrics() {
  resp, err := e.grpcClient.GetBiogasMetrics(ctx)
  if err != nil {
    log.Printf("gRPC error: %v", err)
    return
  }

  gasFlowGauge.Set(resp.GasFlow)
  powerOutputGauge.Set(resp.PowerOutput)
  temperatureGauge.WithLabelValues(
    resp.Sensor
  ).Set(resp.Temperature)
}
```

Monitoring Benefits

- Real-time biogas performance, custom alerts

[1] https://github.com/clementnuss/saia-pcd-exporter

Biogas Plant Dashboard



Real-time biogas plant monitoring and analytics

Electric Fence Control

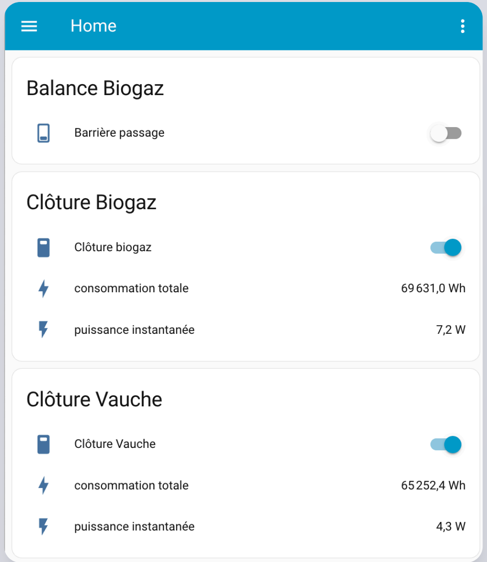
Remote Fence Management

- Shelly MQTT relays for power control
- Kubernetes-hosted MQTT cluster
- Home Assistant integration

Use Cases

- Emergency fence shutdown / repairs

Shelly 1PM Gen3 MQTT relay



Balance Biogaz

Barrière passage

Clôture Biogaz

Clôture biogaz

consommation totale 69 631,0 Wh

puissance instantanée 7,2 W

Clôture Vauche

Clôture Vauche

consommation totale 65 252,4 Wh

puissance instantanée 4,3 W

Home Assistant fence control

🔌 **Shelly → MQTT ← Home Assistant**

Milk Vending Machine

Self-Service Farm Shop

- Fresh eggs, flour, sausages, and milk

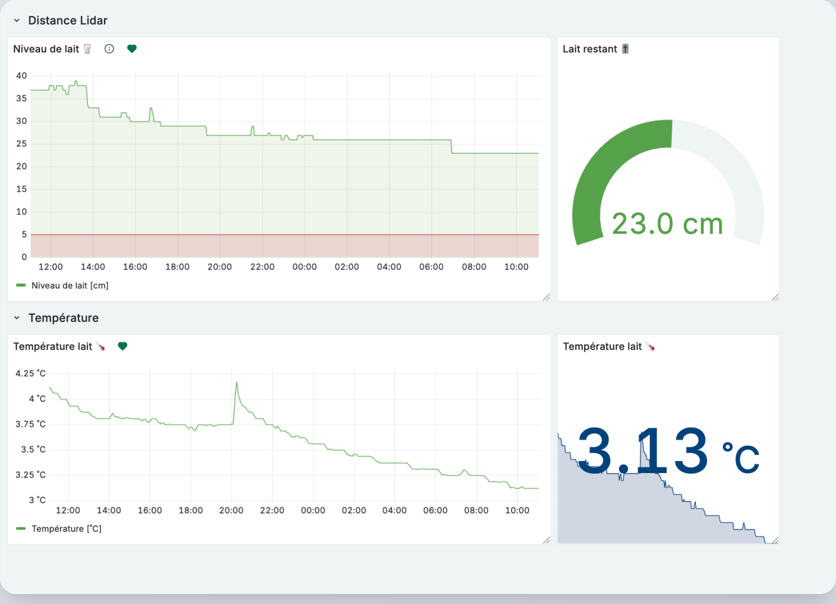
LiDAR Integration Challenge

- Monitor milk levels in the 40L tank
- Automated alerts when machine is empty

Tank Geometry: 1cm height = 1 liter 🧮

$$1L = 10^{-3}m^3 = \pi \cdot r^2 \cdot 10^{-2}m$$

$$r = \sqrt{\frac{10^{-1}}{\pi}} = 0.178m = 17.8cm$$



Milk vending machine monitoring dashboard

Farm Invoicing System

InvoiceNinja[1] - Open Source Invoicing

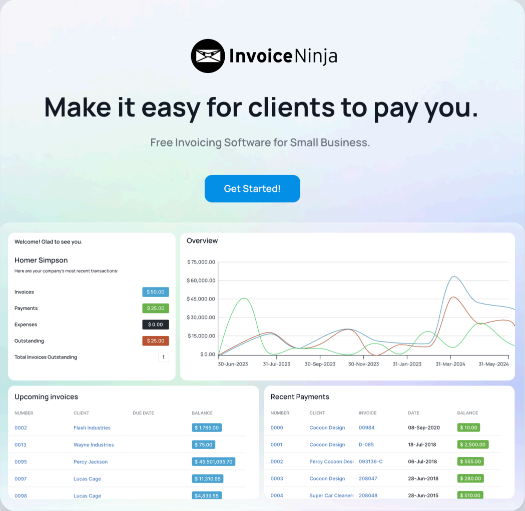
- Complete invoicing and billing solution
- Customer management and payment tracking

Farm Use Cases

- Silage work for third parties
- Hay sales and crop drying

Cloud-Native Deployment

- MariaDB Operator[2] → HA MariaDB
- InvoiceNinja Helm Chart



InvoiceNinja

Make it easy for clients to pay you.

Free Invoicing Software for Small Business.

Welcome! Glad to see you.

Homer Simpson

Here are your company's most recent transactions:

Invoices	$ 50.00
Payments	$ 25.00
Expenses	$ 0.00
Outstanding	$ 25.00
Total Invoices Outstanding	1

Overview

Upcoming invoices

NUMBER	CLIENT	DUE DATE	BALANCE
0002	Flash Industries		$ 1,765.00
0013	Wayne Industries		$ 75.00
0095	Percy Jackson		$ 45,501,095.70
0097	Lucas Cage		$ 11,310.65
0098	Lucas Cage		$4,839.55

Recent Payments

NUMBER	CLIENT	INVOICE	DATE	BALANCE
0000	Cocoon Design	00984	08-Sep-2020	$ 10.00
0001	Cocoon Design	D-085	18-Jul-2018	$ 2,500.00
0002	Percy Cocoon Desi	093136-C	06-Jul-2018	$ 555.00
0003	Cocoon Design	208047	28-Jun-2018	$ 280.00
0004	Super Car Cleaner:	208048	28-Jun-2015	$ 510.00

InvoiceNinja dashboard

1. https://invoiceninja.com/
2. https://github.com/mariadb-operator/mariadb-operator

Live Demo

Real Farm Data

- Current milk production
- Biogas plant performance
- Electric fence status

Security Considerations

- One IT guy 👨‍💻
- Managing electric fences ⚡
- Remotely controlling livestock barriers 🐄
- What could possibly go wrong? 🦕 🦖

Jurassic Park[1]

[1] https://www.jurassicsystems.com/

Lessons Learned

✅ What Worked

- On-premises cost savings
- Talos Linux stability
- GitOps workflow

🔄 Challenges

- Legacy system integration
- Reverse engineering

Thank You! 🙏

Questions?

Cloud-native principles work everywhere, even on Swiss farms! 🐄

The farm that started it all

clement.n8r.ch

in ⌂

> **From Kubernetes to Cows -**
> **Happy to discuss both! 🚀 🐄**