KDD for Enaccess: PAYG command, IoT Data and Visualization platform options

Decision Context

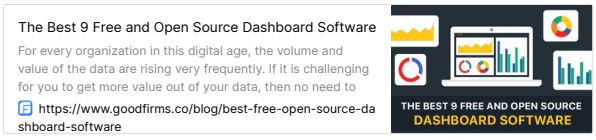
PAYG devices fall under the IoT 'edge device' umbrella. A small set of SaaS companies are currently offering integrated loan management and device management services to PAYG-IoT distributors, primarily focused on token-based GSM home-system devices with some extensions for data feedback. Companies like Angaza have proprietary hardware+software IoT stacks that offer API-integration at some levels while companies like Solaris offer more open source codebases for device firmware and token software. Neither offer an open-source provisioning or analytics platform, although both have SaaS offerings for analytics and PAYG control, integrated with their custom loan platform. Both have a per-end-user-per-month revenue model which is consistent with contemporary SaaS - their revenues scale directly with their customer's customer bases.

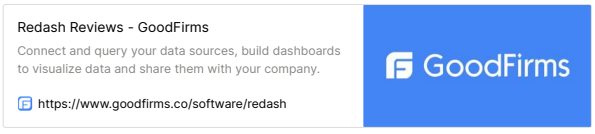
would like to build or buy a PAYG + IoT Data system that has API integration, configurable analytics, cost-effective implementation, ability to serve partners as well as ability to aggregate data over BLE / GSM / Keypad / Wired systems. EnAccess has provided a grant to build an open-source multi-medium IoT communication protocol and data format to support these goals in 2021, with a focus on a deliverable that can enable upstart companies in this area to easily overcome the PAYG-IoT technology barrier.

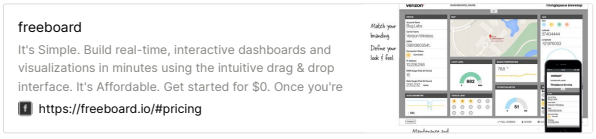
We need to decide on the best alternative approach to building this IoT database and related tools. managed and no-code tools enable speed of secure and scalable business process automation with low overhead costs. We believe this approach has long term value and hence we give priority to options which have managed or no-code cores. We consider two high-level options
a. custom database with open source dashboarding tools (no-code device/partner management can follow alternate no-code/code analysis)
b. open source IoT platforms

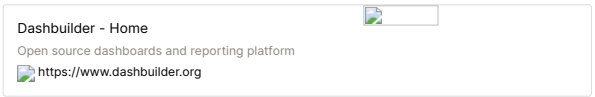
a. Custom db + Dashboard Options

(IaaS, e.g. managed Redis on DigitalOcean + custom droplet with code)









Pros of a custom db+dashboard approach

Freedom to adopt managed or self-managed databases without lock-in

Completely custom server code i.e. process triggers and PAYG responses

The communication layer ends at the database, cleanly separating the application layer which can be full-custom

Cons of a custom db+dashboard approach

Requires coding competence to pre-process incoming IoT data stream

Requires domain-expert skill for building of device provisioning and basic analytics flows

A central database managed by one entity e.g. EnAccess would only require to handle communication layer while application layer would be handed off, making the central db more easily viable compared to a solution with an application layer

Requires a full-time administrator to manage IoT connection to rest of business apps platform

New workflow features require coding and hence take weeks to develop

Initial adoption by business takes longer due to coding requirements

b. IoT Platform Options

(PaaS if managed or IaaS+PaaS in case of self-managed, includes closed-source options for reference)

Guide to IOT Dashboards and Platforms
IOT dashboards are essential for visualizing IOT data, and controlling IOT devices. A Dashboard is a crucial component of any IOT platform. An IOT platform is an
http://www.steves-internet-guide.com/iot-mqtt-dashboards/

dweet™ - Bug Labs Enteprise IoT Platform
Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skateboard dolor brunch. Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf moon tempor, sunt aliqua put a bird on it squid single-origin
https://dweetpro.io/pricing.html

IoT Analytics - ThingSpeak Internet of Things
ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak
https://thingspeak.com

SQL Database for IoT & Sensor Data | CrateDB
"CrateDB is the only database that gives us the speed, scalability and ease of use to collect and aggregate measurements from hundreds of thousands of industrial
https://crate.io/use-cases/iot-sensor-data/

CRATE.IO

ThingsBoard - Open-source IoT Platform
ThingsBoard Read more ThingsBoard Read more ThingsBoard Read more ThingsBoard allows you to create rich IoT Dashboards for data visualization and
https://thingsboard.io

ThingsBoard

Particle Company News and Updates | Particle
Particle is a fully-integrated IoT platform that offers everything you need to deploy an IoT product.
https://particle.io

Homepage - Thingstream by u-blox IoT Communication-as-a-Service
u-blox IoT Communication-as-a-Service makes it simple to get data from IoT devices to the enterprise. Accelerate your IoT.
https://thingstream.io

Learn More - ThingSpeak IoT
This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you
https://thingspeak.com/pages/commercial_learn_more

➤ Kaa Demo | Kaa IoT Platform
We use cookies and other tracking technologies to understand how you interact with our website by collecting and reporting information anonymously.
https://www.kaaproject.org/demo

itead/IoTgo
IoTgo is an open source IoT platform, like WordPress, ZenCart and all other open source software, you can deploy your own IoTgo cloud service. We at ITEAD Studio
https://github.com/itead/IoTgo

OpenRemote | The 100% Open Source IoT Platform
Our ambition is to overcome the challenges of integrating many different devices and data sources into a single, easy to use asset and data management solution. With
https://openremote.io

Introduction to the Azure Internet of Things (IoT)
The Azure Internet of Things (IoT) is a collection of Microsoft-managed cloud services that connect, monitor, and control billions of IoT assets. In simpler terms, an IoT
https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-introduction

Google Cloud IoT - Fully Managed IoT Services
Benefits Accelerate business agility with IoT data from your physical devices Boost intelligence from edge to cloud Google Cloud IoT services gather real-time
https://cloud.google.com/solutions/iot

AWS IoT - Amazon Web Services
There are billions of devices in homes, factories, oil wells, hospitals, cars, and thousands of other places. With the proliferation of devices, you increasingly need solutions
https://aws.amazon.com/iot/

aws

Pros of an open-source IoT Platform approach

Leverages a pre-built best-practices approach to managing incoming IoT data-stream

Leverages a device provisioning and basic analytics platform that is ready to go, reducing the startup-building burden

New workflows can be built quickly as many of these platforms offer drag and drop UIs for process triggers based on incoming IoT connections

Companies can choose a managed or SaaS model for the same service if their business model supports that choice better than a self-managed PaaS

Cons of an open-source IoT Platform approach

Adopters would be initially oblivious to implementation details before they can study the large amount of platform source code in the specific programming language

The Application layer comes with presumptions about IoT management that may not apply across all businesses

Any central entity such as EnAccess who manages a common db might need to provide application-level client-management/API as well as database management and communication layer level API

Perspectives on Approach

[redacted] experienced the often hidden time-cost and domain-knowledge complexity of building device-provisioning/onboarding flows for IoT systems, and considers provisioning an important complement to the IoT data/PAYG flow when considering approaches facilitating new ventures in this field. Standardizing this while considering privacy best-practices could reduce a big barrier, further abstracting away the technical details for integrating PAYG IoT with other business applications.

Data Retention management and Analytics is another natural feature desired of GSM IoT collections. We conjecture that most IoT analysis usually pivots on a single plotted variable for a particular device class e.g. power used by time of day for energy products, along with some standard status variables e.g. location, error state. PAYG control also has common requirements e.g. on/off control or use-metered control.

Hence there are opportunities to design a platform that has pre-built, privacy-enabled standard features for device provisioning, single-variable control and single-variable graphing with map and status indicators and a built in retention policy. Such a platform could enable adopters of the project to incorporate standard IoT outcomes easily into their business operations.

Lock-in risks as well as the central role of EnAccess in enabling upstarts points to the importance of open source, modular approaches that allow the scaling of individual components as managed or self-managed entities, such as front end databases and servers that run data processing code.

Culling the options

AWS/Azure/IBM IoT offerings were not considered the right cost-value tradeoff due to the complexity of adoption and the fact that the dataset of most adopters of this project will be limited in size and will manage with smaller IT teams i.e. not tens of millions of devices/interactions per day managed by a specialist IT team, but hundreds of thousands at most managed by a multi-tasking IT team. The caveat is potentially losing out on AI integration which could be useful for predictive tasks. In the PAYG-IoT business case, learning and prediction requirements are as yet not well defined as business differentiators and hence AI was not considered a prime factor.

The alternatives list was further limited by the following parameter choices: Open source, Free/freemium versions and no per-device fees. Per-devices services are roughly $2/month/device (in addition to any network/SIM card fees), which adds up quickly when selling a large number of smart devices and can be margin-limiting in low-cost markets. This consideration discounted dweetpro.io, thingstream.io, particle.io and thingspeak.com

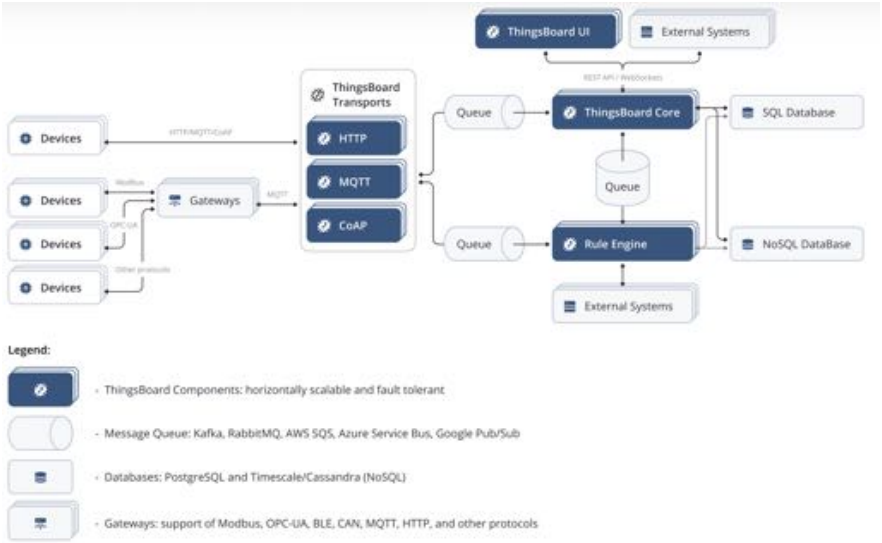
Last Round Alternatives

Feature	custom development	ThingsBoard.io ✓	Kaaproject.org	OpenRemote.io ✗	Crate.io ✗
Total Cost of Ownership for paid premium option - 1 year, 10,000 devices	$120/yr code server, $180/yr Redis managed, $180/yr postgresql managed = $480/yr	$3600/yr for Docker+managed db or $6,000/yr for SaaS cloud, provides flexibility	$3600/yr for self-managed or $24,000/yr for managed on-premise	$3600/yr for self-managed, no cloud offering	db-only $2616/month ✗
Model	IaaS: DigitalOcean droplet with custom stream-processing code, managed Redis db buffer for incoming data providing high-rel front end, then managed PostGreSQL db for IoT metrics. Custom device/partner management workflows on another platform e.g. Airtable-like no-code platform	Open Source or Managed PaaS: Monolithic/microservice interchangeable, IoT stream processing, device/partner management and and IoT Analytics platform in Docker on a DigitalOcean droplet with Aiven-Cassandra + PostGreSQL managed dbs on DigitalOcean. One downloadable package architecture, can be clustered	Managed PaaS: microservices type IoT stream processing, device/partner management and and IoT Analytics platform in Docker connecting to a Redis + PostGreSQL. No monolithic download so steeper initial learning curve	Assissted deployment only, no managed cloud options	Open Source or Managed PaaS: CrateDB is a distributed SQL database built on top of a NoSQL foundation. Customers often use CrateDB to store and query machine data. This is because CrateDB makes it easy and economical to handle the velocity, volume, and diversity of machine and log data.
Open source	—	Yes, updated in 2021	Yes, updated in 2021	Yes, updated in 2021	Yes, updated in 2021
Performance / scale	Full-custom	Single-container / managed cluster / cloud SaaS optionality	Kubernetes baked		Highly scalable
Lock-in risk	Nil	Company could decide to fork free/paid codebases. Timestream data could be migrated but tenant management structure would probably need to fork codebase	Timestream data could be migrated but tenant management structure would probably need to fork codebase	Timestream data could be migrated but tenant management structure would probably need to fork codebase	Developing for specific db type could lead to harder migration options
Basic Data External forwarding and Workflow triggers	—	Yes, Kafka stream and Rulechain configured from drag and drop UI	Yes Redis based queue	'Rule-chain' for PAYG response seems very basic without custom scripts, only math/text functions ✗	Needs external SQL queries but performance is fast enough to support datastream analysis
Vendor info / any risk to business	SLA is fully dependent on internal developers	thingsboard pro used by Engie, no SLA for community edition, dependent on internal developers	FDA and HIPAA as customers	Germany focus with some cities adopting it also Schipol security	—
Feature dev speed	~1wk-1mo	~1day-1wk (most common features are UI based)	~1day-1wk (most common features are UI based)	~1day-1wk (most common features are UI based)	~1wk-1mo

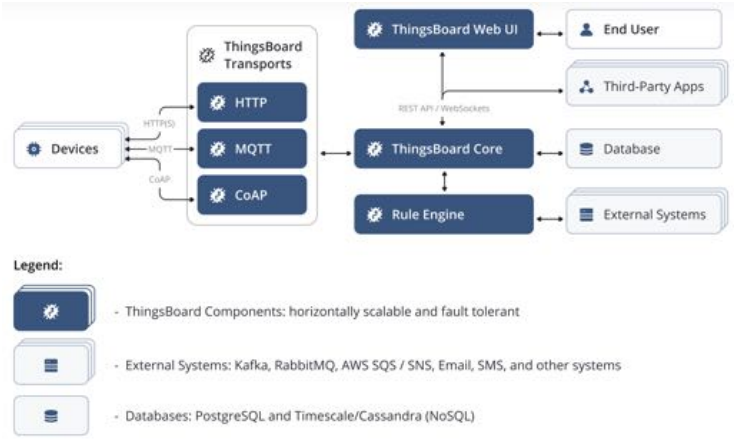
Feature	custom development	ThingsBoard.io ✓	Kaaproject.org	OpenRemote.io ✗	Crate.io ✗
IoT Comms	Customizable	POST/MQTT/CoAP upload, GET request or Gateway subscribing via MQTT	POST/MQTT/CoAP upload, GET request or Gateway subscribing via MQTT		—
API	Will need to be built	REST with JWT auth per device/entity	REST API		—
Analytics for metrics	No	Yes	Yes		Not inbuilt ✗
Device management	Will need to be built	Yes with profiles and gateway MQTT-only devices	Yes with profiles, versions and gateway MQTT-only devices		No ✗
Data management possiblity	Managed Redis + Managed PostgreSQL	Part of docker image as starting point, Managed Cassandra (Aiven on DigitalOcean) + PostgreSQL (native DigitalOcean) database	Managed Redis + Managed PostgreSQL		
Customer management	Will need to be built	Yes with groups and embeddable client dashboards	Yes with groups		No ✗
Customer support	Internal	Only for cloud otherwise internal	Only for cloud otherwise internal		
Interoperable format compatibility	Yes	JSON key-value pairs, additional Customizable 'connectors' for custom binary	JSON		
Chart options	Will need external service e.g. No code big-data service	Moving line/bar/speed-gauge/map	Moving line/bar/speed-gauge/map		No ✗
Filters on displayed charts	—	Only time filters?	Only time filters?		No ✗
Dashboard UX	—	Basic, sufficient	Basic, sufficient		—
Login based filtering	Custom development	Yes	Yes		
SSO types for users/customers	Custom development	Oauth2	Oauth2		
Programming language	PHP	Java	Go	Go	Java
Device/Sandbox codebase	—	not seen any	Arduino samples, web sandbox, etc		Python client example
Git / source Link	—	https://thingsboard.io/docs/user-guide/install/digital-ocean/ also https://github.com/thingsboard/thingsboard	https://github.com/kaaproject/kaa	https://github.com/openremote	https://github.com/crate

Proposed Solution: thingsboard.io docker monolithic

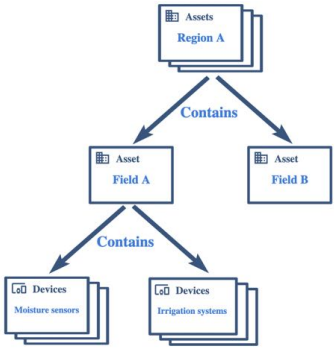
Scale Architecture



Deployment proposal: Lock-in mitigation is by using a managed database for Cassandra/PostgreSQL

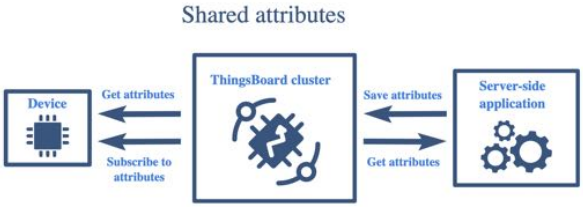


Multi-vendor management from a single central db

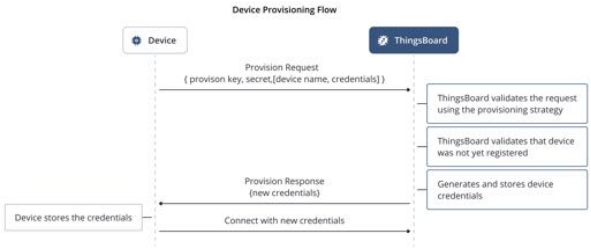


Private devices and dashboards via tenancy

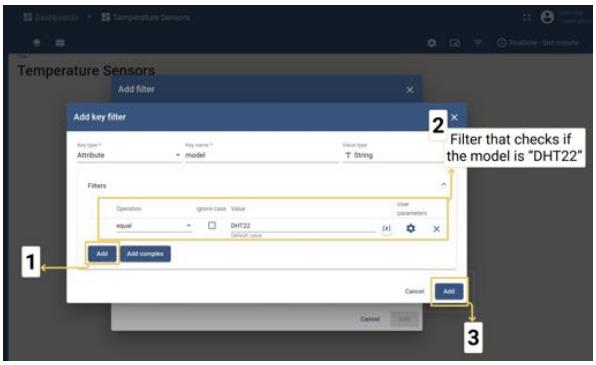
PAYG control



Self-Provisioning



Plotting the primary variable by device type



Feasible Implementation Plan

- Setup DigitalOcean Ubuntu droplet
- Setup Thingsboard.io docker image
- Test custom protocol integration, write connector if required
  - Test VPN integration with Aeris
- Buy managed PostGreSQL db on DigitalOcean
  - Reconfigure Thingsboard configuration-database connection
- Buy managed Cassandra droplet on DigitalOcean from Aiven
  - Reconfigure Thingsboard timeseries-database connection
- Setup administration for EnAccess and tenancy for Simusolar and Tulima Solar
  - Setup tenant profile including dashboard template

    - Setup provisioning flow on Simusolar servers to attach to Simusolar tenant
- Configure phone app to act as MQTT gateway for protocol-compliant devices, including claiming flow