

# Getting Started with the Climate Adaptation Data Platform

Brian Lee Yung Rowe

2024-06-05



# Contents

<b>1 Preface</b>	<b>5</b>
<b>2 License</b>	<b>7</b>
<b>3 Overview</b>	<b>9</b>
<b>4 Quick Start</b>	<b>11</b>
4.1 System Requirements . . . . .	11
4.2 Building the Platform . . . . .	12
4.3 Platform Demos . . . . .	13
4.4 Viewing the dashboard . . . . .	13
<b>5 Use Cases</b>	<b>15</b>
5.1 Public health . . . . .	16
5.2 Precision agriculture . . . . .	22
5.3 Hotter Times . . . . .	26
<b>6 Physical Infrastructure</b>	<b>29</b>
6.1 Direct WiFi connections . . . . .	31
6.2 BLE connections . . . . .	31
<b>7 Mobile app</b>	<b>33</b>
<b>8 Device Operation</b>	<b>35</b>
8.1 Components . . . . .	35
<b>9 CADP Architecture</b>	<b>37</b>
9.1 Design principles . . . . .	37
9.2 Database . . . . .	39
9.3 Sensor data ingestion . . . . .	39
9.4 Weather forecasting . . . . .	40
9.5 Early warning system . . . . .	40
<b>10 Forecasting algorithms</b>	<b>41</b>

10.1 Air temperature . . . . .	41
10.2 Soil moisture . . . . .	41
<b>11 Risk Protocols</b>	<b>43</b>
11.1 Soil moisture . . . . .	43
<b>12 References</b>	<b>45</b>

# Chapter 1

## Preface

We are witnessing the effects of climate change quickly accelerate. The consequences of these changes are just starting to be felt. To avoid widespread suffering, we need to similarly accelerate climate adaptation initiatives.

The Climate Adaptation Data Platform (CADP) aims to do just that. By bringing together multiple stakeholders, we can solve multiple climate-related challenges concurrently. This drives down costs and increases reach.

Consistent with the above goal, the CADP is an open source project. This guide details how to use and develop the CADP. Those wishing to contribute to the CADP should read this guide first. Those who only want to use the CADP can skip most of the theory and focus on the latter half of the book.



# **Chapter 2**

## **License**

Copyright (C) 2024 Zato Novo, LLC.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.



# Chapter 3

## Overview

The Climate Adaptation Data Platform (CADP) accelerates climate adaptation initiatives by bringing together multiple stakeholders to addresss multiple climate-related challenges. The primary stakeholders are policymakers/governments, farmers, and individual members of the community. The CADP provides tangible value to each stakeholder.

The magic of the CADP is that it breaks through policy paralysis by creating the device and data infrastructure that policymakers need to drive widespread climate adaptation efforts. Our hypothesis is that it's unrealistic for municipalities to deploy and operate sensor networks. It's also too complex an undertaking for community groups to take on.

CADP provides an alternate path where individuals and businesses purchase devices that incrementally build out the infrastructure needed for policymakers.

This document discusses the different use cases that CADP is designed for and the value people get from these use cases. Where applicable, a distinction will be made between the value individuals get from the platform versus the value policymakers get.

After describing the use cases, we move into more technical territory so people can learn how to use and develop the platform further. First is a high-level view of the physical infrastructure and the relationship between a device network and the CADP. Devices are responsible for monitoring environmental conditions and sending the collected data to the platform.



Figure 3.1: A prototype weather station device deployed at the University of Dar es Salaam test farm in Tanzania.

# Chapter 4

## Quick Start

The Climate Adaptation Data Platform is a turnkey platform for building weather-dependent applications. CADP accelerate food security and public health initiatives by providing all the infrastructure to automatically incorporate hyperlocal forecasts as a data source for

- precision agriculture
- digital advisory services
- heat-health early warning systems
- infectious disease early warning systems

### 4.1 System Requirements

#### 4.1.1 Linux

The platform will work on most Debian-based Linux distributions. At a minimum you need to have the following available:

- bash: this is the command line shell where you can execute commands
- make: this is a standard tool to build software

In a new environment, run the `bin/init_workstation.sh` script to ensure you have the tools you need to build the system.

##### 4.1.1.1 Windows users

If your primary operating system is Windows, you will need to install the Windows Subsystem for Linux. This will give you `bash` and likely `make`.

#### 4.1.1.2 Mac users

OS X is built on BSD. Many of the tools have slight differences from GNU Linux. That said, the system *should* run without issue on Mac OS X.

#### 4.1.2 Docker

Docker is required. The version used to create this repo is

```
$ docker --version
Docker version 25.0.3, build 4debf41
```

If you are on Windows, follow the official instructions for installing the Linux Subsystem for Windows and Docker desktop.

## 4.2 Building the Platform

The full platform can be built using the following command:

```
make all
```

This will first build the platform and then launch it. The following tasks are executed:

- initialize database
- initialize the dashboard
- build all images

You can use this command repeatedly. The `Makefile` is smart enough to know not to re-initialize the database.

To verify that the system is running, you can check the logs or check the container statuses directly. Logs can be monitored using `make logs`, while the container statuses can be viewed by typing `docker ps`. In this latter command, the output should look something like this:

```
$ docker ps
CONTAINER ID        IMAGE
c86a8b8a34a2      zeomancer_system-weather_forecast
6e2812b24695      zeomancer_system-simulator_etl
27e46eb20065      zeomancer_system-ecm_etl
5db4dd92afa2      timescale/timescaledb:latest-pg12
3bb9e8c4788a      nodered/node-red:latest
dc8c438b1bf3      eclipse-mosquitto:2.0.18-openssl
42c737867360      gcr.io/micro-dynamo-351417/bank-transaction-classification_service
"tini -g ./node-red"
"tini -g ./mosquitto"
"tini -g ./timescaledb"
"node-red"
"/bin/sh -c ./mosquitto_start.sh"
"node-red"
```

To stop the platform, just type `make stop`. You can restart the platform by running `make run`. The previous state will be retained when you start back up.

## 4.3 Platform Demos

A number of demos can be deployed into your running system.

### 4.3.1 Get available demos

This target shows all the available demos and the regions where stations can be loaded.

```
make list-demos
```

### 4.3.2 Deploy a demo

Deploying a demo means that all source weather data is downloaded and loaded into the database. Once the source data are loaded, it is possible to run a simulation to show how the system state evolves over time or batch run all the forecasting models and risk rules until an arbitrary end date. The drawback of the simulator approach is that it will take a while to observe.

```
make run-demo
```

During the installation of a demo, you will be prompted to choose whether to continue with running batch forecasts or not.

### 4.3.3 View dashboard

There are two dashboard views. One defines the data processing workflows and is for application developers. The other is for policymakers and shows the device network and corresponding weather and application information.

### 4.3.4 Reset system

To run another demo, you need to reset the system state. A `make` target makes this a simple task. Be careful: it also makes it easy to lose all your work, since it deletes everything in the database!!

```
make reset
```

## 4.4 Viewing the dashboard

### 4.4.1 Policymaker dashboard

The end-user reporting dashboard for decisionmakers and policymakers is located at <http://localhost:9002/dashboard/>.

#### **4.4.2 System administration dashboard**

The workflows that define the processing jobs are defined in Node RED. They are accessible at <http://localhost:9002/#flow/>.

# Chapter 5

## Use Cases

Every device powers multiple simultaneous use cases. We call this use case stacking. This is different from re-using or re-purposing a device. With use case stacking, multiple applications run concurrently. The supported use cases depend on the sensors attached. The two main public health use cases only need the base set of sensors, which means anybody with an account can get hyperlocal heat risk and mosquito abundance information.

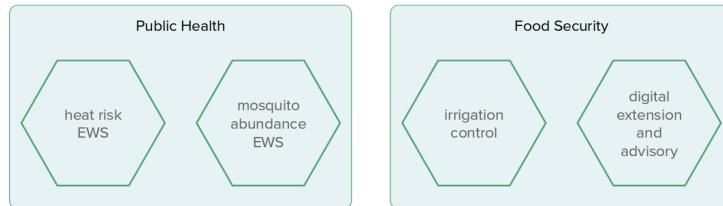


Figure 5.1: Use case stacking enables the CADP to power multiple simultaneous applications across public health and food security.

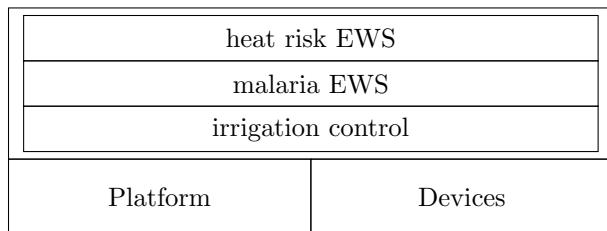


Figure 5.2: The CADP provides a common infrastructure that powers multiple concurrent use cases.

## 5.1 Public health

### 5.1.1 Early warning systems

The genesis of this project was the death of 11,000 people in Libya due to a medicane (Mediterranean hurricane) that caused a dam to fail. People were caught off guard because as a failed state, Libya has no functioning weather agency. It turns out that most of Africa has limited weather stations. Most official weather agencies only send out seasonal forecasts.

Weather forecasts give people critical time to plan and prepare for major weather events. Building on top of weather forecasts are early warning systems (EWS). These systems apply rules, or risk protocols, to forecasts to determine the level of risk people are exposed to. EWSes can operate on any type of forecast and includes heat risk, air quality risk, the spread of wildfire, the spread of infectious disease, and the spread of pests like the ash borer.

The UN declared a mandate that everyone should be covered by an early warning system by 2030. Some \$3 billion dollars have been earmarked for this effort. While many countries have early warning systems, many are outdated, no longer work, or cover a limited number of risks.

The CADP provides the infrastructure to quickly build EWSes and fulfill the UN's vision.

### 5.1.2 Heat risk

Survivability is the upper limit of temperature and humidity that humans can withstand. Humidity has a significant impact on the maximum temperature that humans can survive in. With high humidity, the body loses its ability to regulate temperature via perspiration.

In some places, climate change is making heat waves more extreme and pushing temperatures to the survivability level. Even if that limit isn't reached, whether those conditions are livable is another matter. Livability limits help people understand what activities are possible given temperature and humidity (and sun exposure). Figure 5.3 compares livability limits for two population groups and whether the activity takes place in direct sun or shade.

The conversion formula between METs ( $M$ ) and calories is

$$M \frac{3.5m}{200} = kcal/min, \quad (5.1)$$

where  $m$  is mass in kilograms. This formula enables us to customize the heat risk of specific activities according to each individual and their specific health conditions.

Heat risk builds on the concept of livability. The goal is to provide an EWS that helps people understand their heat risk exposure and make decisions about what

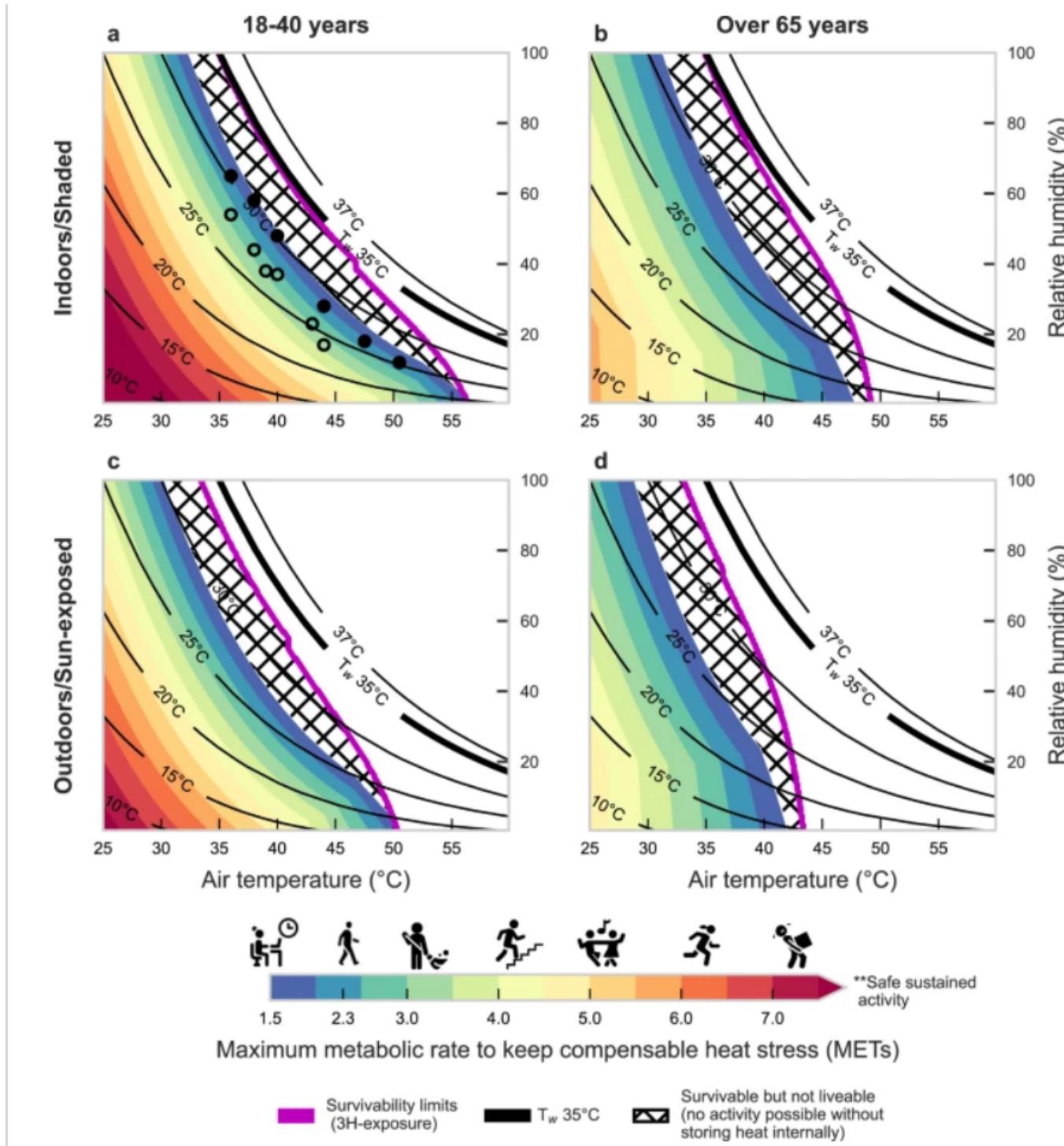


Figure 5.3: Livability limits for different METs at different temperature and humidity levels. Source: [@vanos2023]

activities to do or how to mitigate heat risk during certain activities. As shown in 5.4 the platform currently highlights risks

Individuals can get personalized heat risk guidance for any device in CADP. Alerts can be delivered via email or via the mobile app. Users can enter some health information to get better alerts. This information is saved but segregated from personally identifiable information (PII).

Relevant health information includes

- age
- BMI (weight + height)
- diabetes
- heart issues

We envision a conversational alert message that doubles as a daily planner. Integrating this information with preferred activities results in specific messaging that informs people of what activities are realistic for a given day. On a very hot and humid day, an older person may want to limit sun exposure to an hour. That might mean that golf should be avoided, whereas a stroll through a shaded park is more appropriate.

### 5.1.3 Mosquito-borne infectious disease

Mosquitos are a common disease vector and can carry numerous infectious diseases, such as malaria, dengue fever, zika virus, yellow fever. Disease outbreaks are driven by the growth of a mosquito population. To limit the severity of an outbreak, mosquito surveillance is conducted to monitor the spread of mosquitos. By tracking mosquito abundance, it is possible to predict how large an outbreak might be.

Direct observation of mosquitos is an involved process. It requires specialized devices to attract and trap mosquitos. Usually these devices need periodic cleaning.

A simpler approach is to use a mosquito abundance model and build risk protocols on that model. These can be calibrated with physical traps (future). Figure 5.5 shows a system of partial differential equations (PDEs) that describe the change in mosquito abundance given a few parameters. Of importance is that both temperature and humidity drive the population growth of mosquitos.

We can thus use the temperature and humidity forecasts available in the CADP to make mosquito abundance forecasts. Figure 5.6 shows a graph of a device in the CADP. The green line represents observational data, while the orange time series is the forecast produced by the CADP.

1. Install device in observation area
2. Enable mosquito abundance alerts

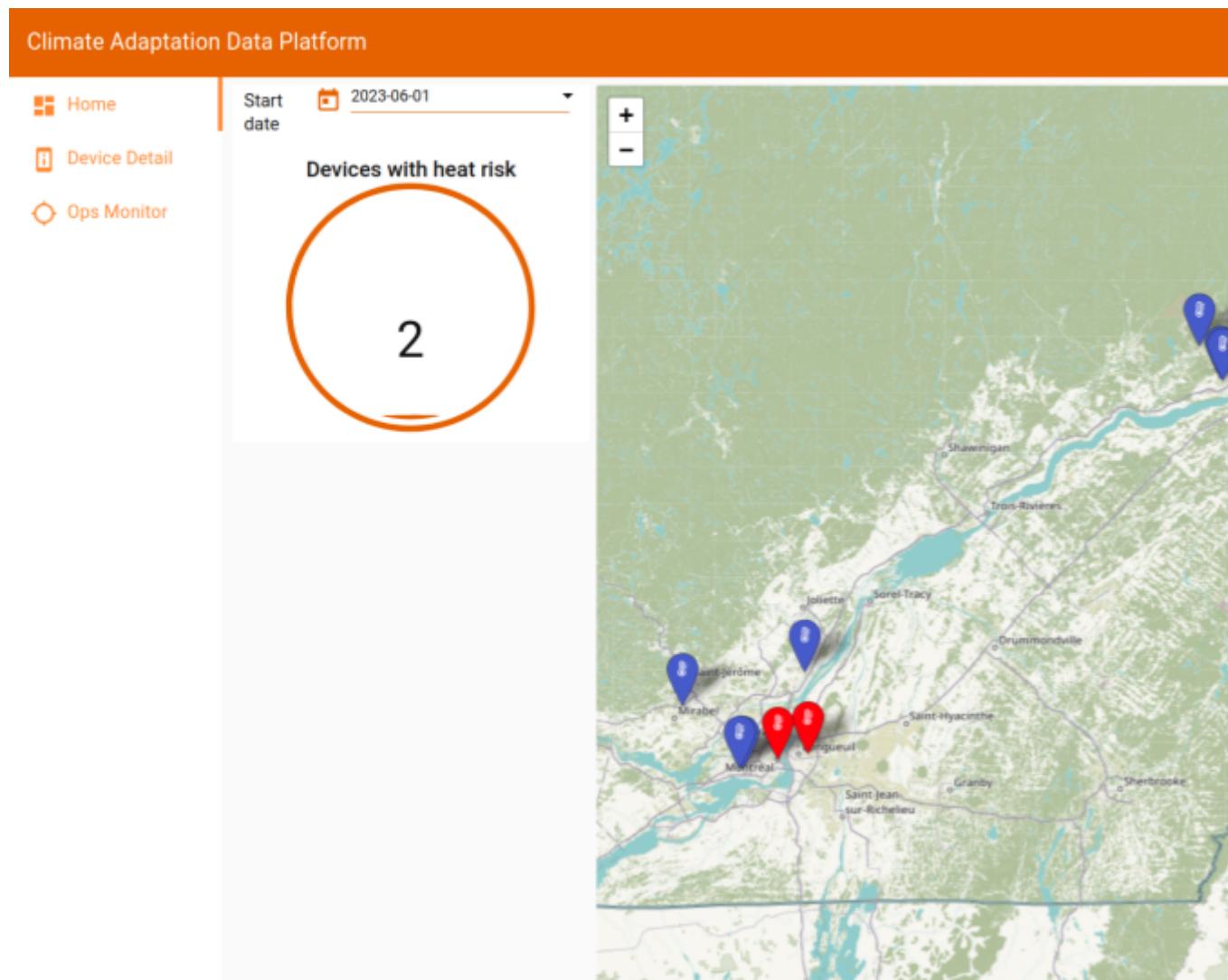


Figure 5.4: A screenshot of a policy dashboard showing active heat risk alerts for two devices.

$$\begin{aligned}
\frac{dE}{dt} &= \beta A_o - (f_E + \mu_E) E \\
\frac{dL}{dt} &= f_E E - \mu_L \left( L + \frac{L}{k_L} \right) - f_L L \\
\frac{dP}{dt} &= f_L L - (f_P + \mu_P) P \\
\frac{dA_{em}}{dt} &= Pf_P \sigma e^{\mu_{A_{em}} \left( 1 + \frac{P}{k_P} \right)} - (\mu_{A_{em}} + \mu_r) A_{em} + \dots \\
\frac{dA_b}{dt} &= f_{A_{em}} A_{em} + f_{A_o} A_o - (\mu_A + \mu_r) A_b - f_{A_b} A_b \\
\frac{dA_g}{dt} &= f_{A_b} A_b - \mu_A A_g - f_{A_g} A_g \\
\frac{dA_o}{dt} &= f_{A_g} A_g - (\mu_A + \mu_r) A_o - f_{A_o} A_o
\end{aligned}$$

Figure 5.5: A system of partial differential equations that model mosquito population growth based on environmental conditions. Source: [@erraguntla2021]

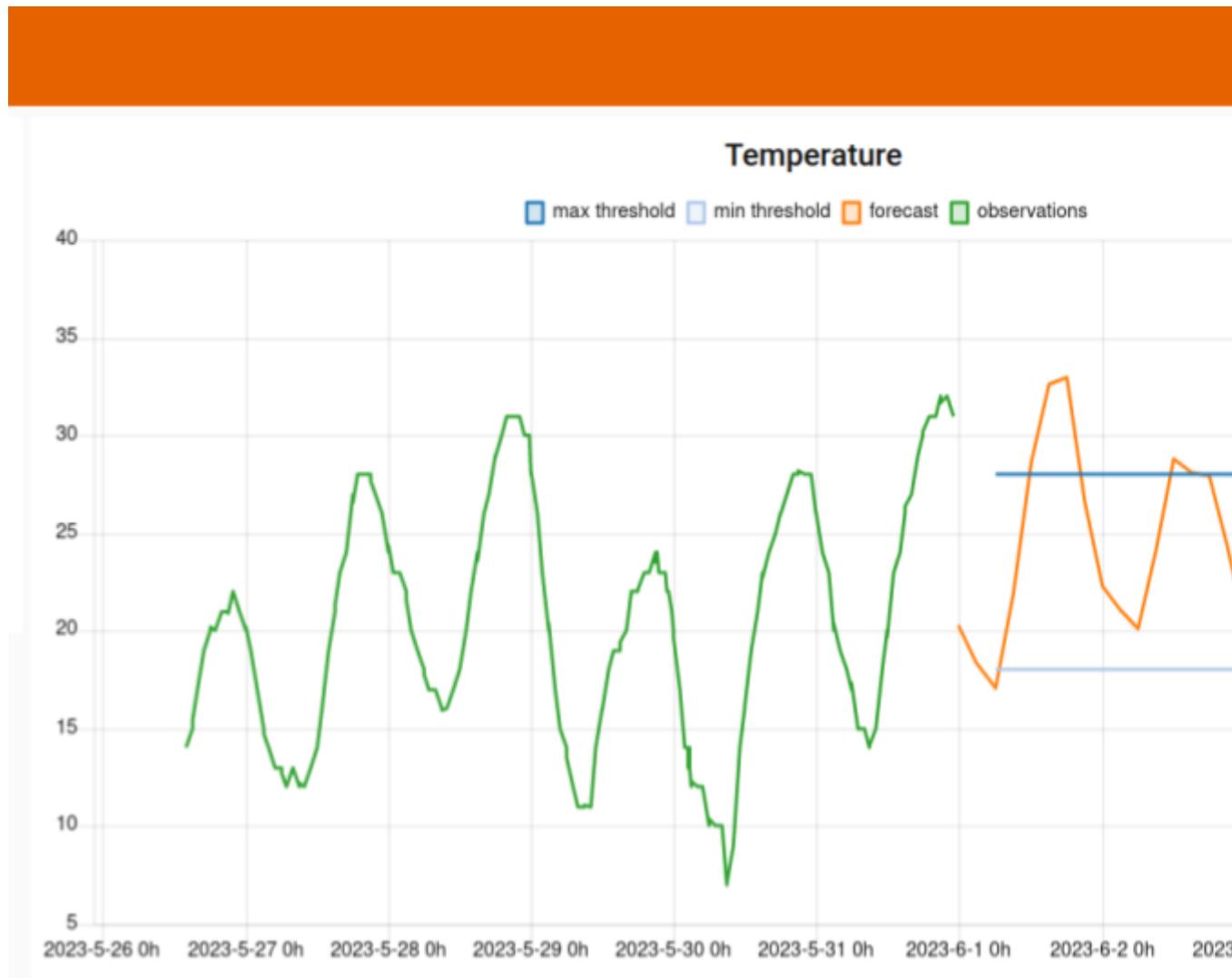


Figure 5.6: A device detail view showing historical temperature, a temperature forecast, plus heat risk thresholds.

## 5.2 Precision agriculture

Helping farmers in Africa was the genesis of this project. The CADP has the potential to accelerate precision agriculture initiatives. Precision agriculture can increase crop yields and protect against drought (food security) by conserving water usage.

Reducing water usage also improves public health. Many areas rely on rain water as their primary potable water source. That means irrigation is often competing with personal hydration for water. Optimizing water usage means there can be sufficient water for both needs.

Precision agriculture is a complex field. Our aim is to simplify the process and reduce its cost to make it accessible to smallholder farmers worldwide.

We phase in the approach with the following steps:

- dry soil forecasting
- watering schedules
- crop calendars
- risk protocols
- irrigation control

### 5.2.1 Dry soil forecasting

Forecasting dry soil is the first step in irrigation control. Soil that is either too dry or too wet can be harmful to plants. While the optimal amount of water is specific to each plant, preventing over and underwatering can be accomplished simply by monitoring soil moisture.

A soil moisture forecast can notify the farmer or gardener when soil needs watering. By waiting for a watering signal, this approach avoids overwatering, which conserves water.

The deployment model shown in Figure 5.8 is relatively simple. A single sensor device is required that has a soil moisture sensor attached to it.

### 5.2.2 Crop watering needs

Watering depends on the type of plant growing. Some plants need a lot of water while others don't. Many plants don't like wet roots, so watering cannot be too frequent. We need to develop these control models for each plant.

Water needs change based on the maturity of the plant. Beans need moist soil when pods are forming. To form large roots, beets need minimal water during early growth stages to promote root development.

We can extract this data from various agricultural sites, such as the Farmers Almanac. The information needs to be encoded into programmatic rules.

Crop calendars show the planting season and duration for different regions.



Figure 5.7: UDSM professors and agronomy staff standing behind a prototype weather station device.

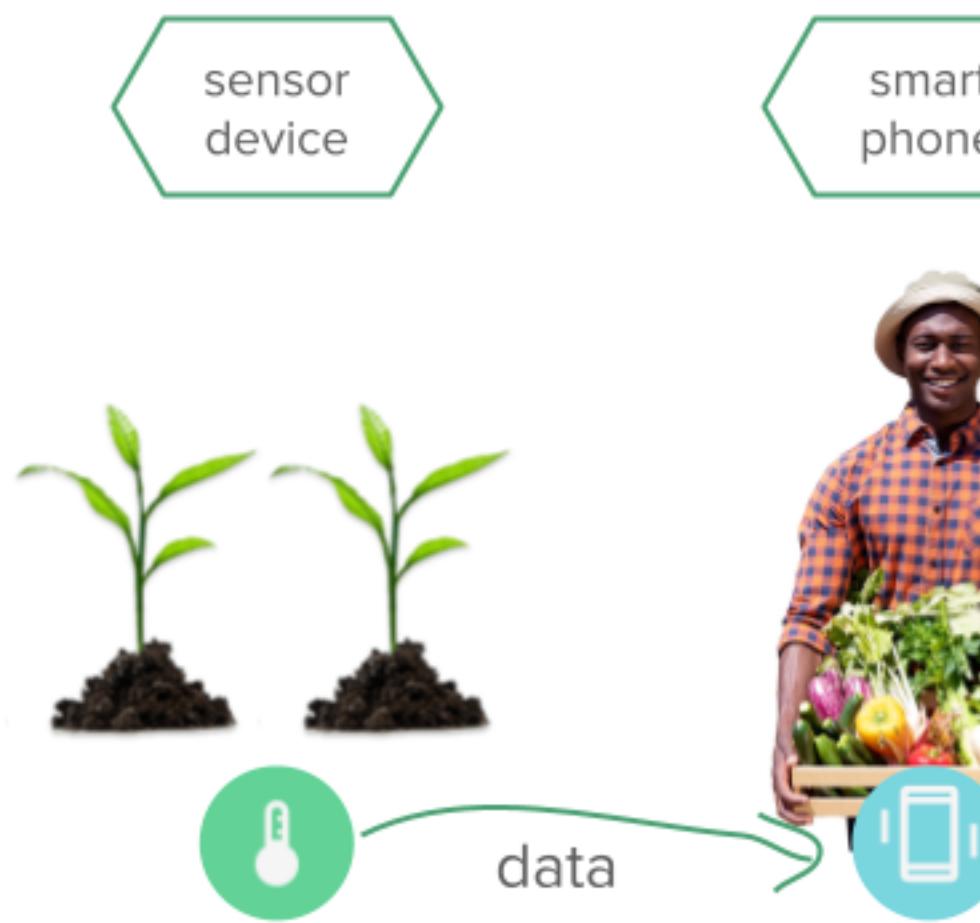


Figure 5.8: A simple installation where a single device monitors soil moisture for a crop.

## Kenya – Crop Calendar

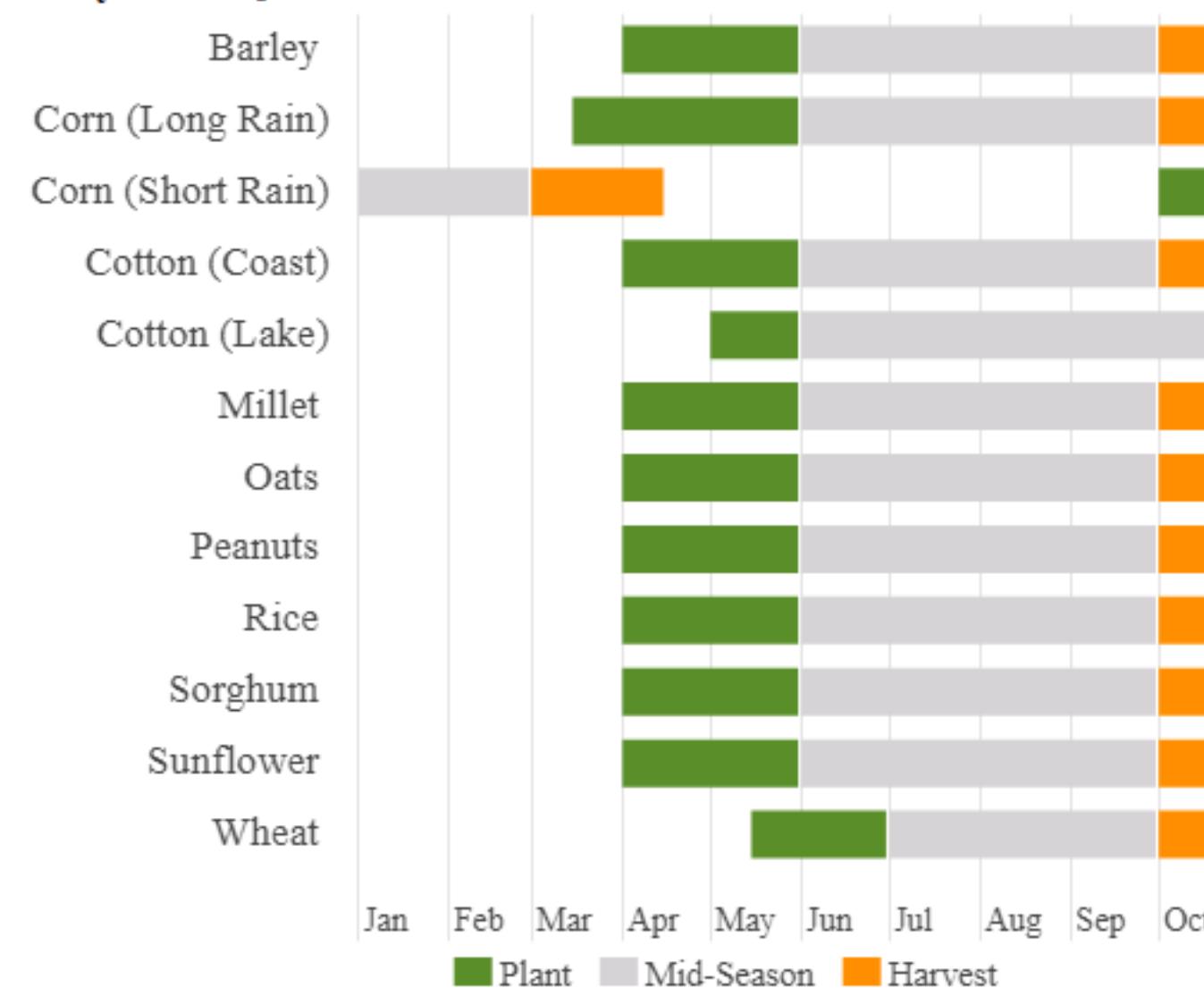


Figure 5.9: Crop calendars for Kenya. Source: [@usda1]

We use the approach defined by Brouwer and Heibloem (1986) that determines water need based on evapotranspiration  $ET_o$  and a crop factor  $K_c$ .

TODO: Incorporate information from this webpage: <https://www.fao.org/4/s2022e/s2022e07.htm#3.2.4%20determination%20of%20crop%20factors>

### 5.2.3 Risk protocol for dry soil

Knowing whether soil needs irrigation depends on a few factors. First, the crop determines the general watering schedule. Decisions are made based on reconciling watering needs with current environmental conditions. We need to know the current soil moisture level. We also want to know the future soil moisture level, which factors in weather forecasts. There's no point watering today if it will rain tomorrow.

### 5.2.4 Irrigation control

Automated irrigation control builds on top of the water needs forecasting model.

### 5.2.5 Freeze warnings

Farmers and gardeners in temperate or similar climates need to know when there are frosts that could damage fragile seedlings. Uncertainty leads to wasted time preparing for non-events or stress worrying about whether plants will be okay.

Crop tables typically show temperature ranges that specific crops can tolerate.

## 5.3 Hotter Times

This site is used to display public data related to the sensor network. Hotter Times thus becomes a public service where everyone can benefit from the device network.

All of the layers discussed above can be displayed on Hotter Times.

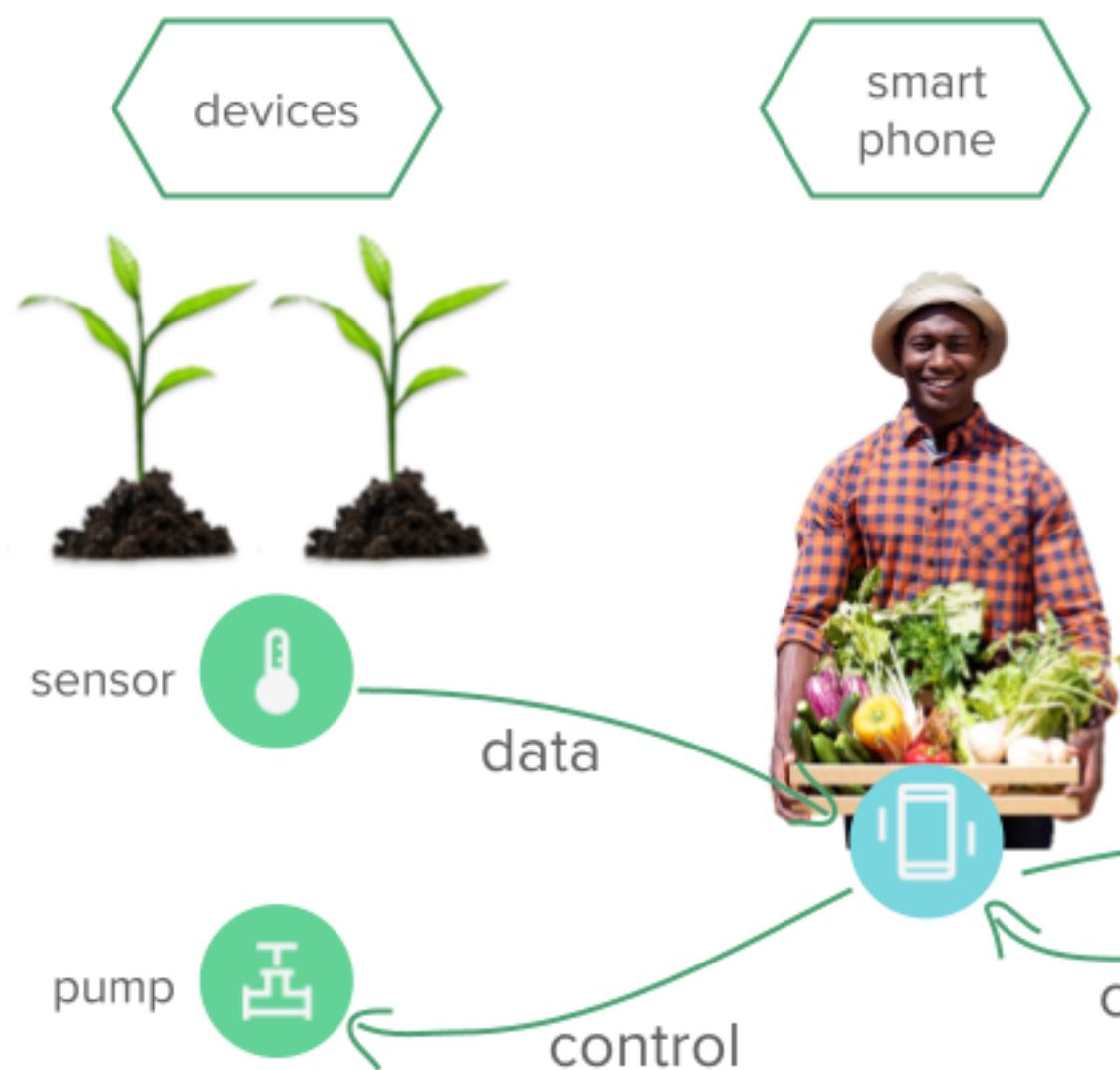


Figure 5.10: A more sophisticated system with a separate device that controls an irrigation pump.

Vegetable Crop	Frost Tolerance	Start Weeks Before Spring Set out Date	Set Out Weeks Before Avg. Frost Free Date	Set Out Weeks After Avg. Frost Free Date	Minimum Soil Temp.	Preferred Air Temp.	Length of Season in days	Start Weeks Before Fall Frost Date*	Zodiac Signs
Broccoli	hardy	4-6	4	2-3	40 F	60 F	55-85	15	Cancer, Scorpio, Pisces, Libra
Cabbage	hardy	4-6	5	2-3	20 F	60 F	60-110	15	Cancer, Scorpio, Pisces, Libra, Taurus
Collards	hardy	4-6	4	2	40 F	60-65 F	70-80	15	Cancer, Scorpio, Pisces
Kale	hardy	4-6	5	2	40 F	60-65 F	65-75	15	Cancer, Scorpio, Pisces
Onion (seed)	hardy	4-6	6	2	40 F	55-75 F	92-130	20	Libra, Taurus, Pisces, Cancer, Sagittarius
Peas	hardy	n/a	4	2-3	40-55 F	60-65 F	55-75	16	Cancer, Scorpio, Pisces, Taurus
Spinach	hardy	4-6	3-6	4	40 F	35-75 F	60-65	8	Cancer, Scorpio, Pisces
Turnips	hardy	4	4	2	50 F	60-65 F	35-50	13	Cancer, Scorpio, Pisces, Taurus, Capricorn, Libra
Beets	half-hardy	n/a	4	2-4	40 F	60-65 F	55-80	14	Cancer, Scorpio, Pisces, Capricorn, Libra
Carrots	half-hardy	n/a	4	2-4	45-85 F	60-65 F	60-80	15	Taurus, Cancer, Scorpio, Pisces, Libra
Cauliflower	half-hardy	2-4	4	2	60 F	60-65 F	55-110	15	Cancer, Scorpio, Pisces, Libra
Celery	half-hardy	2-4	4	2	60 F	60-65 F	90-140	n/a	Cancer, Scorpio, Pisces
Lettuce	half-hardy	2-6	2	3	40 F	60-65 F	45-100	13	Cancer, Scorpio, Pisces, Libra, Taurus
Mustard	half-hardy	2-4	5	2	45 F	60-65 F	30-50	15	Cancer, Scorpio, Pisces
Potatoes	half-hardy	n/a	2-4	2-4	40 F	60-65 F	100-115	16	Cancer, Scorpio, Pisces, Taurus, Capricorn, Libra
Radishes	half-hardy	2-4	4-6	2-4	40 F	60-65 F	25-50	8	Cancer, Scorpio, Pisces, Taurus, Capricorn, Libra
Parsnips	half-hardy	2-4	4	3-4	50 F	40-70 F	80-120	15	Taurus, Capricorn, Cancer, Scorpio
Swiss Chard	half-hardy	2-4	3-4	4	50 F	40-85 F	50-60	15	Cancer, Scorpio, Pisces
Beans, snap	tender	n/a	n/a	1-2	60 F	70-80 F	40-55	15	Cancer, Scorpio, Pisces, Taurus, Libra
Beans, shelling	tender	n/a	n/a	1-2	60 F	70-80 F	85-115	15	Cancer, Pisces
Corn	tender	n/a	n/a	2-3	60 F	60-75 F	70-100	16	Cancer, Scorpio, Pisces
Okra	tender	n/a	n/a	3-4	60 F	70-85 F	50-60	16	Cancer, Scorpio, Pisces, Libra
Tomatoes	tender	6-10	n/a	4	60 F	70-85 F	65-85	15	Cancer, Scorpio, Pisces, Libra
Cucumber	very tender	2-3	1	2	60 F	65-85 F	50-75	16	Cancer, Scorpio, Pisces
Eggplant	very tender	9-10	n/a	2-3	65 F	70-85 F	65-80	15	Cancer, Scorpio, Pisces, Libra
Luffa	very tender	4-6	n/a	2-4	70-95 F	65-75 F	120	18	Cancer, Scorpio, Pisces
Melon	very tender	2-4	n/a	2-3	50 F	80-100 F	80-125	16	Cancer, Scorpio, Pisces
Peppers	very tender	6-8	n/a	2-3	60 F	55-90 F	60-80	16	Cancer, Scorpio, Pisces
Pumpkins	very tender	3	n/a	2-3	40 F	65-75 F	90-120	16	Cancer, Scorpio, Pisces, Libra
Squash, summer	very tender	4	n/a	3-4	60 F	65-75 F	45-75	16	Cancer, Scorpio, Pisces, Libra
Squash, winter	very tender	4	n/a	3-4	60 F	65-75 F	70-120	18	Cancer, Scorpio, Pisces, Libra
Sweet Potatoes	very tender	6-8	n/a	2-3	65 F	70-85 F	150-170	18	Cancer, Scorpio, Pisces, Taurus, Capricorn

Figure 5.11: Planting guides for different vegetables

## Chapter 6

# Physical Infrastructure

Devices can connect with the CADP via different means. Figure 6.1 shows two different approaches. The simplest approach is to connect directly via WiFi. In this approach, a device communicates directly with the CADP MQTT server. This approach works well when reliable WiFi is available.

The method used for remote areas with limited infrastructure utilizes a mobile phone as an intermediary. In this approach we assume that mobile phones (with a network connection) exist in proximity to devices but they are no fixed internet gateways. We call the mobile phones ephemeral gateways because they may disappear at any time.

Ephemeral gateways (EGs) offer a temporary network connection. The EG is responsible for pushing data to the CADP MQTT server. If a network connection is unavailable, it will cache the data on the mobile phone until a network connection is available. Once a connection is established, the EG must push the data to the server.

Devices are connected together via the LoRa wireless protocol. When a particular device connects with an ephemeral gateway, it broadcasts a message to other connected devices and notifies them that it has a network connection. This device will attempt to upload as much data to the mobile device as possible.

Devices that receive this message will begin broadcasting data to be received by the device connected to the EG. The broadcast includes the ID of the device connected to the EG. Only the device connected to the EG can broadcast an ack back.

In the event that multiple EGs are available, each device is responsible for choosing the device to use as the target EG.

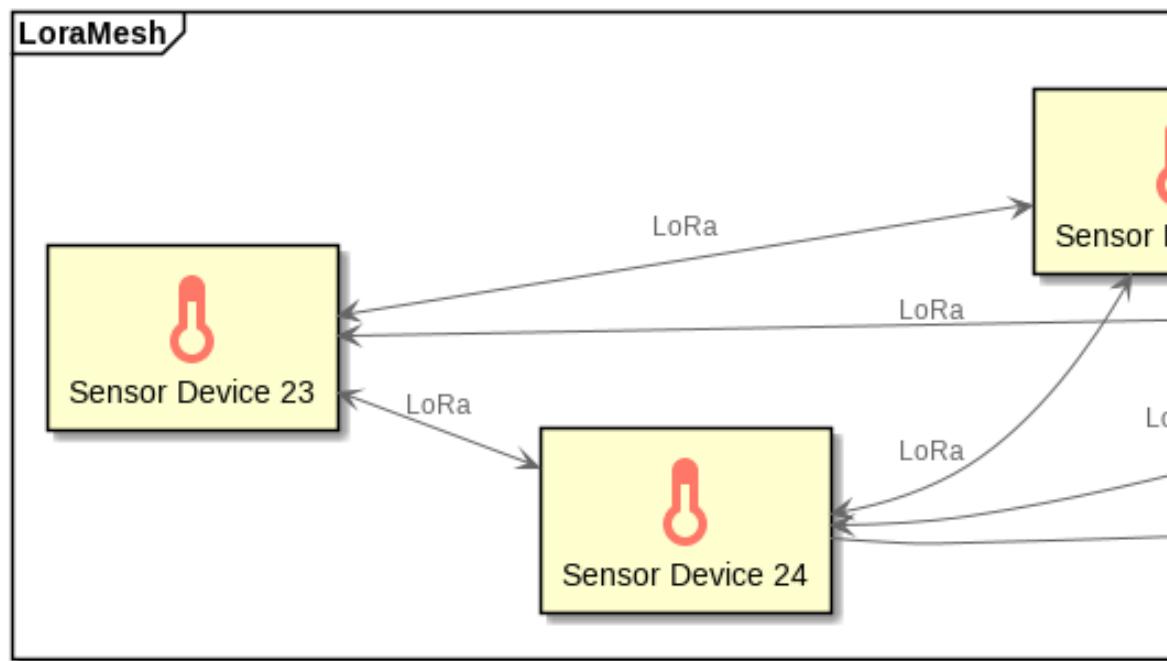


Figure 6.1: Different network configuration of system.

## 6.1 Direct WiFi connections

Devices connected via WiFi will start a MQTT client to send directly to the cloud MQTT broker. By default, the MQTT broker host is `mqtt.zeomancer.com`.

## 6.2 BLE connections

When a smartphone connects to a device via BLE, a few things happen:

- the device sends a test data packet to confirm that data is sent to the MQTT broker
- the device waits for an ack from the smartphone
- if successful,
  - LoRa is activated
  - a `internet_available` message is broadcast over LoRa

### 6.2.1 Data routing algorithm

The data routing algorithm solves one key problem, which is the existence of ephemeral gateways. Devices are connected to each other via LoRa. The goal is to efficiently route data through the LoRa network to devices that are connected to the network (a “connected device”). LoRa works via a broadcast mechanism, so devices are either in range or out of range of a connected device.

If a device is in range, then the device simply broadcasts its data to the target connected device. Any other device that receives the message will ignore the message if the target id does not match its device id.

For a device that is out of range, it will send data to the nearest device in range of a connected device. The devices can be modeled as a graph where each device is a node and edges represent the signal strength between them. We can define two relevant distance metrics between two arbitrary nodes:

- the number of edges (aka hops) between nodes;
- the harmonic mean of the signal strength of the edges separating the nodes.

A device chooses a target device by considering the harmonic mean of the signal strengths between it and a connected device. Generally, the device should choose the path with the highest value. Note that this may change as EGs can move around and disappear altogether.



## **Chapter 7**

# **Mobile app**

A mobile application is used to mediate the network connection for devices in remote areas. It is also used to deliver alerts to users. Using the device as an alert delivery mechanism ensures users have the device installed and able to act as a data transport for devices.



## Chapter 8

# Device Operation

Devices are designed to work with minimal configuration. Even if you have a fixed WiFi connection available, it is simplest to configure the device via your smartphone.

The basic steps are:

- install device in activity area
- pair device via Bluetooth
- configure device via app
  - set location
  - (optional) configure and enable WiFi

### 8.1 Components

The following information is provided in the event that repairs need to be made to a device.

#### 8.1.1 Connectors

- Molex 51005-2P connector for LiPo batteries



# Chapter 9

## CADP Architecture

The architecture of the CADP comprises:

- a low power and long range sensor devices;
- a data collection and storage facility;
- a pluggable predictive models for producing forecasts and predictions;
- a reporting dashboard for decisionmakers and policymakers; and
- a delivery mechanism to provide alerts and suggested interventions to individual stakeholders.

The system has two primary sources of data. The first is baseline weather forecast data, produced by the European Center for Medium-range Weather Forecasts (ECMWF). This is a global, gridded dataset with 0.4 degree resolution. These forecasts are archived in Amazon S3 and are publicly accessible.

The second dataset comes from the devices themselves. The sensor data is used to tune the gridded forecast according to local conditions.

### 9.1 Design principles

- Plan for 18 month shelf life
- Focus on end-to-end infrastructure
- Stub components where necessary
- Use concurrency where appropriate to minimize wall time
- Minimize data movement
- Minimize cost

Infrastructure deployment for multiple regions

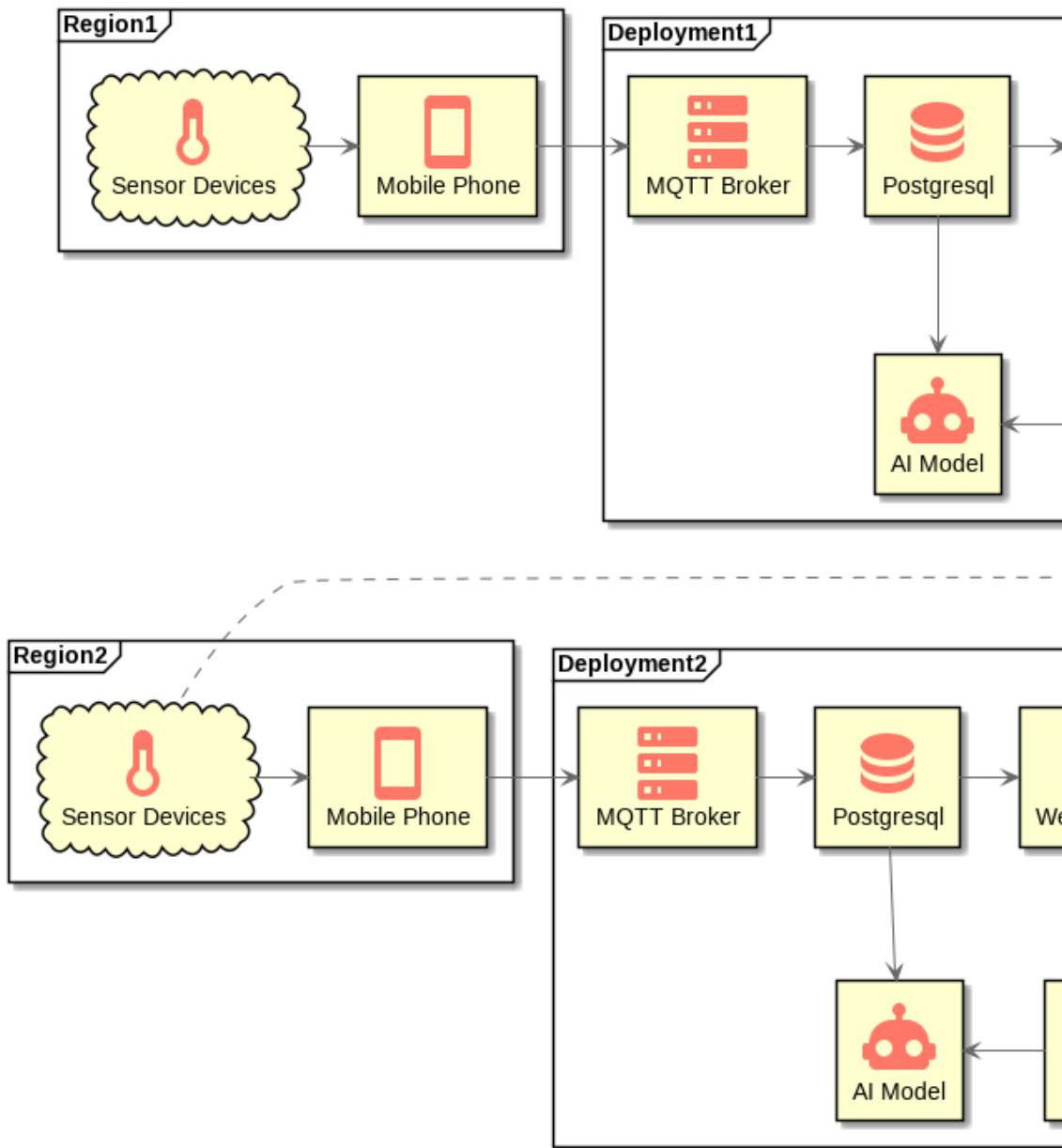


Figure 9.1: System architecture to generate weather forecasts and monitor devices.

## 9.2 Database

### 9.3 Sensor data ingestion

#### 9.3.1 Raw sensor data

Stored as an unnormalized delimited file. These files have no header, and each row is self-contained.

The general format of the file is `key,timestamp,values`, where `values` represents an arbitrary number of additional fields.

For example, the following snippet shows actual sensor data in this format.

```
air_t_h_p,2024-01-15T00:03:26,-5.34,85.30,1048.80
battery,2024-01-15T00:08:23,0.8398108
air_t_h_p,2024-01-15T00:13:20,-5.19,83.84,1049.22
air_t_h_p,2024-01-15T00:23:15,-5.11,82.51,1049.31
battery,2024-01-15T00:28:12,0.8361486
air_t_h_p,2024-01-15T00:33:09,-6.98,83.39,1049.73
air_t_h_p,2024-01-15T00:43:03,-6.97,82.98,1050.29
battery,2024-01-15T00:48:00,0.8359045
air_t_h_p,2024-01-15T00:52:57,-6.99,84.48,1050.52
air_t_h_p,2024-01-15T00:57:44,-5.05,84.28,1050.53
battery,2024-01-15T01:02:44,0.8312657
air_t_h_p,2024-01-15T01:02:45,-6.87,84.85,1050.91
air_t_h_p,2024-01-15T01:12:39,-5.05,84.56,1050.78
air_t_h_p,2024-01-15T01:22:33,-5.06,84.41,1051.12
battery,2024-01-15T01:22:34,0.8307775
air_t_h_p,2024-01-15T01:32:28,-6.99,83.29,1051.41
air_t_h_p,2024-01-15T01:42:22,-6.87,82.33,1051.89
battery,2024-01-15T01:42:22,0.8363928
air_t_h_p,2024-01-15T01:52:16,-6.61,80.60,1052.68
air_t_h_p,2024-01-15T02:01:55,-6.27,80.06,1053.42
battery,2024-01-15T02:01:56,0.831754
air_t_h_p,2024-01-15T02:11:50,-6.34,77.78,1053.46
air_t_h_p,2024-01-15T02:21:44,-6.39,76.10,1053.47
battery,2024-01-15T02:21:45,0.8322423
air_t_h_p,2024-01-15T02:31:39,-6.17,73.03,1053.98
air_t_h_p,2024-01-15T02:41:33,-6.08,71.05,1054.36
battery,2024-01-15T02:41:33,0.8361486
```

Two observation types are included. The first is `air_t_h_p`, which represents air observations. The key hints that there are three value fields. The second observation type is `battery` and contains a single reading for the battery level.

### **9.3.2 Data issues**

When the device cannot connect to the Internet for a while, it loses its internal time, and the clock resets to 2000-01-01. These data points can be dropped.

Even if the clock doesn't reset, there can be clock drift. This can usually be ignored since the observations are aggregated anyway.

## **9.4 Weather forecasting**

The system uses the ECMWF global gridded forecast data as a baseline forecast. Local sensor data is integrated into this forecast to produce a more accurate forecast for specific locations.

### **9.4.1 ECMWF forecasts**

ECMWF forecasts are updated every six hours. The publicly available AWS archive<sup>1</sup> has a complete history of data. Usage for the bash script that downloads the data appears in Appendix A.

There are up to 48 forecast steps per forecast. Each forecast step ranges between 27 MB - 50 MB, which implies a minimum of 1.3 GB of data is retrieved every six hours (per update). To save space, these files need to be slimmed down by removing all unnecessary data.

## **9.5 Early warning system**

---

<sup>1</sup><https://registry.opendata.aws/ecmwf-forecasts/>

# Chapter 10

## Forecasting algorithms

### 10.1 Air temperature

### 10.2 Soil moisture

We want to forecast soil moisture based on current environmental conditions. There are four main variables that likely drive soil moisture:

- air temperature
- air humidity
- precipitation
- current soil moisture

We can use a recursive least squares (RLS) implementation as a starting point. This is the same approach used for the temperature forecast, so it should be straightforward to modify.

Predicting soil moisture is confounded by the fact that a change in moisture can be driven by weather but also human intervention. The latter case will create spikes in soil moisture data. These need to be handled with care to avoid erroneous model behavior.



# Chapter 11

## Risk Protocols

### 11.1 Soil moisture

Forecasting dry soil is the first step in irrigation control. The forecast is simply the moisture content of the soil.

When soil moisture reaches a certain level, an alert needs to be generated. A risk protocol defines the soil moisture level that triggers an alert. The risk protocol is governed by a few factors:

- associated crop
- growth stage of crop

Each plant has its own water needs, so risk protocols need to be plant-specific. We assume the soil moisture sensor is for horticulture or agriculture, so there is a specific crop that the sensor is associated with. This could be an herb, a vegetable, or even an ornamental.

Water needs vary depending on the growth stage of the plant. When beets are young, they are intentionally stressed with less water to promote root development. Beans need plenty of water during pod formation. To simplify the model, we can ignore the growth stage aspect of the plant and pick one value.



## Chapter 12

## References



# Bibliography

Brouwer, C. and Heibloem, M. (1986). *Irrigation Water Management: Irrigation Water Needs*. Food and Agriculture Organization of the United Nations.