Fogget-about-it grow management system

Preparing the groundwork for the next agricultural revolution.

Problem definition:

- Over the last few years, there has been research into the cultivation technique known as “fogponics”, a high-intensity form of agriculture which benefits more from automation than any other.

- This project provides a management and control system for a fleet of fogponics units that can be extended to other use-cases.

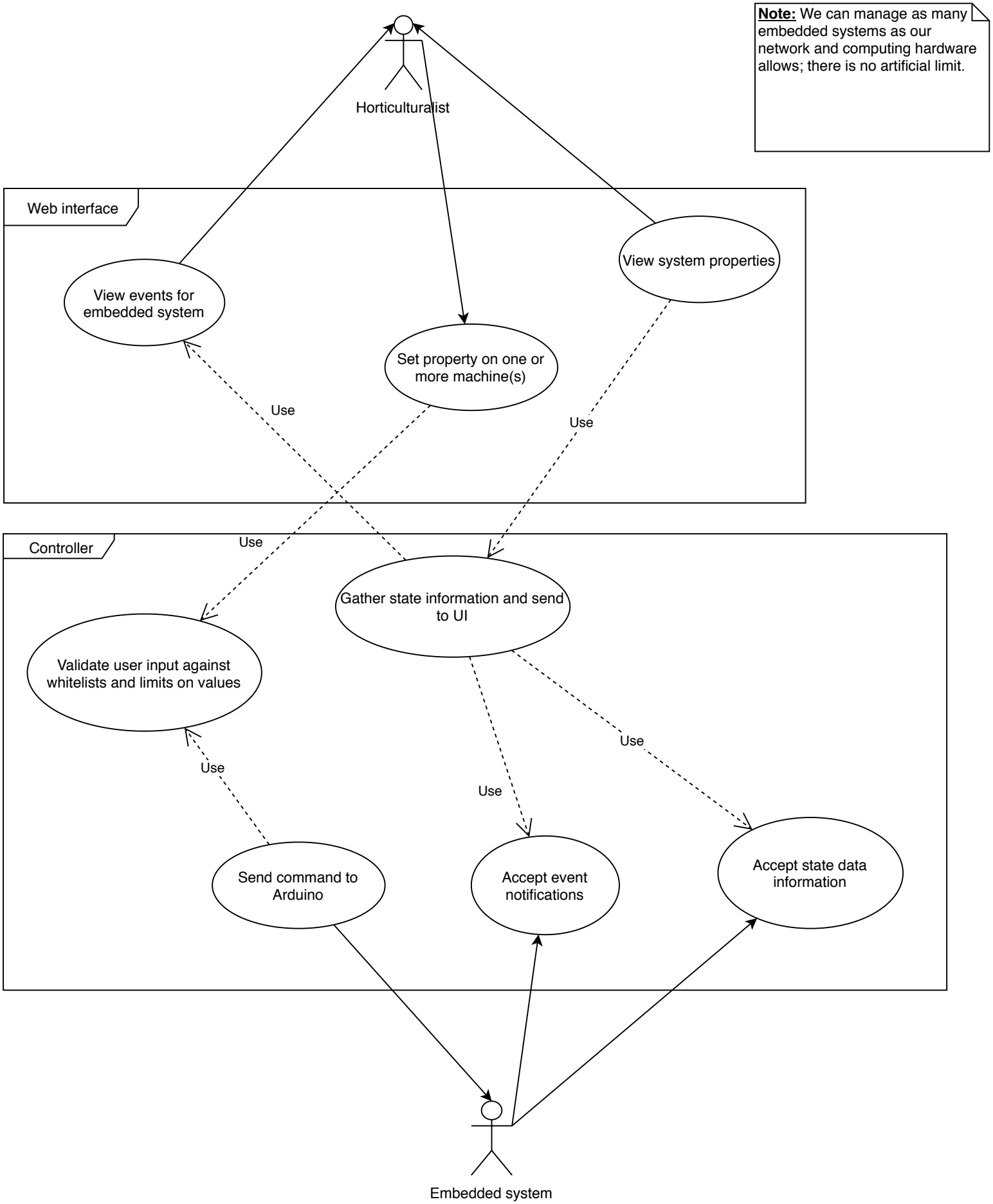
- This is in continuation of a project I have been running off and for half a year. The physical prototype is (mostly) built but lacks a control panel.

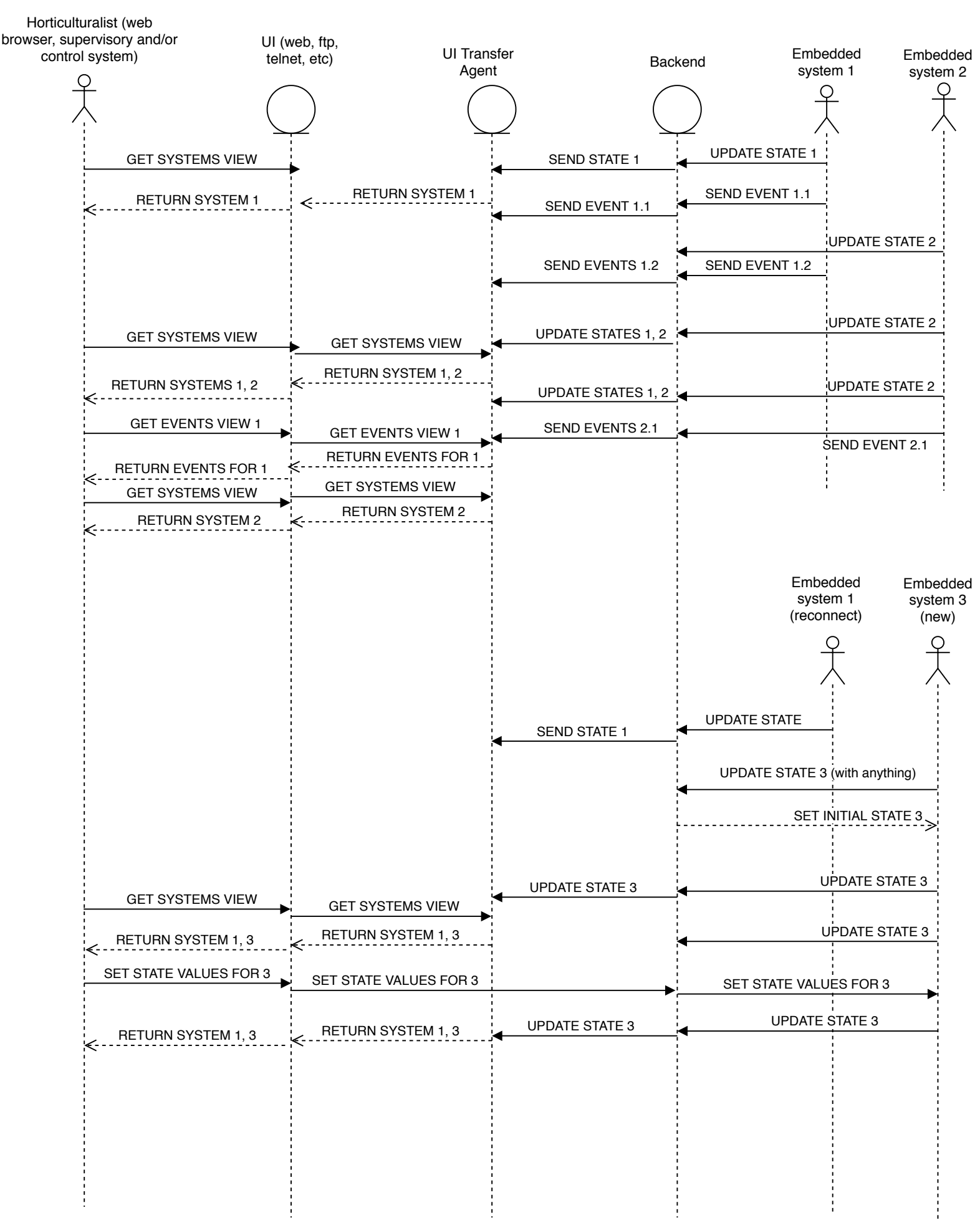
We got ourselves a control panel.

:)

Functional requirements:

- We're building a basic control system for an IoT use-case.
- The control system gathers and updates the current state information of the embedded systems growing plants.
- It must also be able to maintain a list of recent events for perusal by the end-user.
- New systems are to be automatically given sane defaults and added to the system pool. The only user responsibility must be sourcing the embedded system with a unique 64-bit integer ID.
- The end-user must be able to change settings.
- These settings must be validated in order to correct against nonsense inputs.
- "Computer, set the rooting chamber to -273 degrees C…"
- The embedded system running the grow environments should do their its own validation as good engineering, but layers of safety protect against screwups downstream.
- In effect, it acts as an application firewall.
- That's basically it as far as *functional* requirements go.


Note: We can manage as many embedded systems as our network and computing hardware allows; there is no artificial limit.
Horticulturalist
Web interface
View events for embedded system
Set property on one or more machine(s)
View system properties
Use
Use
Use
Controller
Gather state information and send to UI
Validate user input against whitelists and limits on values
Use
Use
Use
Send command to Arduino
Accept event notifications
Accept state data information
Embedded system


Horticulturalist (web browser, supervisory and/or control system)
UI (web, ftp, telnet, etc)
UI Transfer Agent
Backend
Embedded system 1
Embedded system 2
GET SYSTEMS VIEW
SEND STATE 1
UPDATE STATE 1
RETURN SYSTEM 1
RETURN SYSTEM 1
SEND EVENT 1.1
SEND EVENT 1.1
UPDATE STATE 2
SEND EVENTS 1.2
SEND EVENT 1.2
UPDATE STATE 2
GET SYSTEMS VIEW
GET SYSTEMS VIEW
UPDATE STATES 1, 2
RETURN SYSTEM 1, 2
RETURN SYSTEMS 1, 2
UPDATE STATES 1, 2
UPDATE STATE 2
GET EVENTS VIEW 1
GET EVENTS VIEW 1
SEND EVENTS 2.1
SEND EVENT 2.1
RETURN EVENTS FOR 1
RETURN EVENTS FOR 1
GET SYSTEMS VIEW
GET SYSTEMS VIEW
RETURN SYSTEM 2
RETURN SYSTEM 2
Embedded system 1 (reconnect)
Embedded system 3 (new)
UPDATE STATE
SEND STATE 1
UPDATE STATE 3 (with anything)
SET INITIAL STATE 3
UPDATE STATE 3
UPDATE STATE 3
GET SYSTEMS VIEW
GET SYSTEMS VIEW
RETURN SYSTEM 1, 3
RETURN SYSTEM 1, 3
UPDATE STATE 3
SET STATE VALUES FOR 3
SET STATE VALUES FOR 3
SET STATE VALUES FOR 3
UPDATE STATE 3
RETURN SYSTEM 1, 3
RETURN SYSTEM 1, 3
UPDATE STATE 3