Learning Network

We want to set up a low-cost local learning network. For this we run a Learning Management System (=LMS, like Moodle) on a microserver and connect 5-10-20 or 50 raspberry pi clients to work on it. On our network we will also run several micro services, and part of our students' projects will be about adding and improving services to our own learning infrastructure.

On the LMS we run our local version of Open Computer Science Courses, supported with our own framework and resources around it.

Definitions:

Microserver: small software which serves content and webdomains to computers in the network.
Micro Services: A software development technique that arranges an application as a collection of loosely coupled services. E.g. one service for user authentication, one for user management, one for a dropbox-like app, one for the lunch schedule. It makes it easy to work on several projects in parallel, and reduces the responsibilities.

- Services can be implemented using different programming languages, databases, hardware and software environment, depending on what fits best.
- Services are small in size, messaging-enabled, bounded by contexts, autonomously developed, independently deployable, decentralized and built and released with automated processes.

Minimum Viable Product: Instead of building a complete solution over many months, we start with something that is as small as possible launch quickly (minimum), that offers offers enough of a value to actually be useful (viable), and that's in itself finished (product).
Open Courses: Some universities and educators have provided their content for (non-commercial) purposes. High-quality content can be made accessible to learners, and we can build a framework around content to help students approach and understand it.

Minimum Viable Product: Moodlebox.net

We can start with a local LMS network, running on a RaspberryPi.

- Connect to the created Wifi network.
- Access the internet, if the RaspberryPi is attached via Ethernet.
- learn.pip.ac could be a subsite where the moodle site can be accessed.
- External SSD with 1TB of video content can be accessed.
- A splash screen can be customised, when connecting to the network.
- Easy to set up, just load the .img to a microSD card
- HTTPS is now integrated
- Most recent version of Moodle

Limitations:

- Although Moodle is pretty lightweight, it may run into resource problems.
- The wifi Antenna, which is used as a router, is not so strong.
- 10-15 clients is the maximum.

Resources:

- https://moodlebox.net/en/ homepage of
- http://lal.ngo/ doing something like we want in Lebanon

Long-term vision:

The 42 network offers excellent coding education for free, through project-based and peer-based learning. A small staff of 24-30 persons can run a school with 2000-3000 students, an efficiency unknown so far. Projects start with a PDF document and a forum to ask questions and share useful resources. Students work in pairs or larger groups on projects, for a few days or even weeks. Finishing one project leads to unlocking another (set of) projects. The platform is also gamified, you earn experience points from completing challenges and you level up. Five randomly selected students give feedback on projects.

The required infrastructure consists of a huge building with at least 150 working stations with iMac computers, a large server cluster, and space for students. It has spread to 20+ countries within a few years since its creation.

The education there is for free, but requires a large investment in space and hardware. We want to build a scalable network, aimed at developing countries to train skills like problem-solving, team-work and software development. At a fraction of the costs, by using Raspberry Pi's as clients.

If we can create a 3000€ coding lab, with 1000€ for every 5 more computers added, we could:

- Offer high-schools with a low budget a high-quality computer lab for high-quality education.
- Offer youth in camps a meaningful occupation to improve their skills and learn from each other.

The details of the platform:

The raw platform consists of three elements, each with a separate technology stack.

1. Client computers: configured RaspberryPi's with a screen, that are in the network and access the server via Ethernet or a Wifi Router.
2. A server with several services:
    a. A Learning Management System, like Moodle, which hosts courses, discussions, projects etc.

b. User Dashboard, where progress of users, such as experience points, levels, and unlocked challenges are visible. The starting point for the user. It can unlock and access content on the other services

c. Central Authentication Service, where users authenticate via OAuth to all other services in the network.

d. Extensions with student projects.

3. An “information point” - one central system which informs participants about what is going on.

   a. A website with information for external visitors.

   b. An announcement system, which can congratulate students who have leveled up, or achieved certain results.

   c. Hardware like lights and sounds which indicate that something special has happened, maintained by students.

   d. A scheduling system that announces if certain events take place, etc.

   e. It can be extended by students with additional functionality.

Current open questions:

How to host our learning platform in the cloud and on local nodes at the same time?
What are the development stories involved in setting up our MVP architecture?