



# Proposal for CoderDojo community platform

Date: 2nd February 2015

Authors: Damian Beresford & Paul Savage

Directors: Cian Ó Maidín and Richard Rodger  
nearForm limited, Tankfield,Tramore, Waterford, Ireland  
Registered in Ireland: 507692 <http://www.nearform.com> +353-1-5143545

## Brief Summary

This is nearForm's proposal for the implementation of the CoderDojo community platform, it is based on an approach that we have found to generate successful projects. nearForm are open source to our core, it's the lifeblood of our company; from contributing to open source projects to building active vibrant communities around open source projects - two very different things! nearForm staff are actively engaged in two local dojos. We have a real-world understanding of the CoderDojo movement and a deep insight into the requirements of the CoderDojo platform.

The approach we're proposing will combine our understanding of the provided brief with our expertise with best of breed open source tools and technologies to build a future proof Platform for the CoderDojo Foundation. This modular / microservice approach is the best and most appropriate way for the community to make open source contributions to the Platform.

We propose that a beta version of the platform will be available for testing in June 2015, with full roll out in September 2015.

We believe that our combination of a strong team, the best practices (technical and non-technical), CoderDojo community involvement, open source experience, proven methodology and an excellent track record of delivery makes us the right partner to develop the new CoderDojo platform.

## Table of Contents

### BRIEF SUMMARY

### PROPOSED APPROACH

API

NODE.JS

MICROSERVICES

UI

MOBILE

LOCALISATION

MODULES AND PLUGINS

DATABASE

DATA PROTECTION

COPPA/CHILD PROTECTION

DEVELOPMENT/DEPLOYMENT

TESTING

DESIGN

REPORTING

HOSTING

PLATFORM UPTIME AND MONITORING

HOW WE WORK

OPEN SOURCE

COMMUNITY INPUT

### RESPONSE TO FUNCTIONAL REQUIREMENTS

TICKETING

OAuth/LOGIN

THIRD PARTY APIs

FORUMS

MOZILLA OPEN BADGES

ZEN REPRODUCTION

MISC

COMMENTS

### NEARFORM PROFILE

EXPERIENCE AND REFERENCE SITES

PERSONNEL

STATEMENT OF CAPABILITIES

### IMPLEMENTATION PLAN

HIGH LEVEL PROJECT SCHEDULE

### SIGNOFF

## Proposed Approach

We propose building the CoderDojo platform using a modern microservices-based approach, combined with best-of-breed open source tools and technologies, where applicable. Everything we use (tools, libraries and so on) will be open source and freely available. They will also be carefully selected to be compatible with the open source license of the platform itself.

### API

It is our view that an application program interface (API) is crucial to the long-term success of the platform, in terms of both future proofing and the extensibility of the platform via open source contributions from the CoderDojo community. Therefore, the cornerstone of our proposal is the CoderDojo platform API. This API is a REST-based web service that acts as the fulcrum around which the various parts of the platform will be developed.

### Node.js

Node.js will be the development language of the platform, for the following reasons:

- Node.js provides a rapid, cross-platform, low-friction development environment.
- Node.js is server side JavaScript, which is currently one of the most (if not the most) widely used programming languages([techcrunch.com/2012/09/12/javascript-tops-latest-programming-language-popularity-ranking-from-redmonk/](http://techcrunch.com/2012/09/12/javascript-tops-latest-programming-language-popularity-ranking-from-redmonk/)).
- Node.js will make the platform easily accessible to open source contributions from the CoderDojo community, ensuring future maintainability and extensibility.

### Microservices

[Microservices](#) are an approach to developing a single application in the form of a suite of small services. In this approach, each service runs in its own process and communicates using lightweight mechanisms. nearForm has been pioneering microservices development in Node.js for over four years and has successfully delivered over 50 client microservices-based projects.

The main benefits of a microservices based architecture for the CoderDojo platform are as follows:

- **Easy development of new services by the community.** Services are isolated in their own code bases, allowing for easy updates and modifications by the community.
- **Future-proofed services.** Services can be individually upgraded or replaced as required over time.
- **System stability and resilience.** Services can be developed in a modular, plug-in fashion (see below).

## UI

For the user interface (UI), we propose a combination of Angular.js and regular server-side templated content generation. Angular is not search engine optimisation (SEO)-friendly, so it will be used mainly for content creation and editing, while the SEO parts of the platform will be generated using [Express.js](#) views (Express is the most widely used web application framework for Node.js; it is used in about 70% of all node.js web server apps).

## Mobile

nearForm has considerable mobile app development experience, both in terms of mobile app development and mobile websites. We are acutely aware that many users, if not the majority, will access the platform from a mobile device, particularly with regard to the ticketing system; for example, checking in participants at the start of a dojo. We will use [responsive design techniques](#) to ensure that the UI is as accessible and usable as possible on a wide range of devices.

While developing a mobile app is not within the current scope of the platform, we are confident that when we reach that stage, our choice of Angular will mean that a significant volume of front end code can be re-used. When implemented correctly, the majority of Angular controllers can be re-used in a mobile application framework such as [Ionic](#).

## Localisation

In previous similar projects, nearForm has successfully followed best practice in node.js and Express.js, for both localisation and internationalisation. This issue is well known in the Node community and the solutions are flexible and extensible. We will make it as easy as possible for community members to provide new translations and update existing translations. We have also heard great things about [Transifex](#), which is free for open source projects, and would like to investigate using it in the CoderDojo platform.

We have found from experience that localisation is easy to do when it is done properly from the start. It is much more challenging to retrofit localisation into an existing project.

With this in mind, we are proposing that we develop the platform with at least two languages from the start. nearForm has internal staff who can provide translations into Irish, German, Polish and Italian.

## Modules and plugins

On both the client and server side, we will be using a modular/plugin based approach to development. This is facilitated on the server side by Seneca ([senecajs.org](#)), a framework for developing microservices. Seneca allows the backend API to be easily extended via Seneca plugins.

Likewise, on the client UI side, we envisage a modular/plugin approach from the start. For example, each Dojo will have a set of core plugins enabled by default ('ticketing', 'users', 'profiles', and so on), and can selectively add additional plugins as they are developed by the community.

An example of this in action would be if community members from one Dojo wanted to create an Attendance module to help track what kids worked on specifically at each Dojo session they attended. This Attendance module would require two plugins: a Seneca plugin on the server side and an Angular plugin on the client side. When complete and successfully used in their own local Dojo, these modules could be promoted in the community with instructions on how to enable and use them for other Dojos. Over time, these community provided modules may make their way into the core modules for use by all Dojos.

We feel this modular/plugin approach is key to easily extending the platform via community contributions and is crucial to the success of the platform in the long term.

We did consider submitting a separate proposal that did not use this approach, on the basis that it would be a less costly project. However, when we did the math, it doesn't save a huge amount of time or money, and we strongly feel that the benefits of a modular/plugin architecture far outweigh negligible short term cost savings.

## Database

While we can be largely database agnostic - the Seneca framework allows for easy switching of databases via its data store plugins - we find that [Mongo](#) is great for rapid development. We typically use Mongo extensively during the initial development phase when the data schema can change very frequently. However, closer to the end of the development phase we usually switch to a more mature database back end.

Our database of choice is [PostgreSQL](#), a proven enterprise class open source solution. We have a lot of experience of running PostgreSQL at scale; many of our customers bet their business on it.

Nonetheless, we are open to feedback from the CoderDojo Foundation (and/or community) on the choice of database.

## Data Protection

In this project, we are particularly aware that data protection is a highly complex area, given that the CoderDojo platform will be used by a global audience. As a rough rule of thumb, we generally follow the guidelines set out by AWS in this document:

[http://d0.awsstatic.com/whitepapers/compliance/AWS\\_EU\\_Data\\_Protection\\_White\\_paper.pdf](http://d0.awsstatic.com/whitepapers/compliance/AWS_EU_Data_Protection_White_paper.pdf). For more specific concerns, we would typically consult with a law firm who are experts in this area and with whom we have worked in the past.

## COPPA/Child Protection

This is a new area for us and one which we take very seriously. As well as strictly following the COPPA guidelines, we will look to other sources for guidance in this area, notably the following:

- <http://kidblog.org/home/privacy-policy/>
- <http://www.moshimonsters.com/privacy>
- <http://www.clubpenguin.com/safety>
- <http://www.clubpenguin.com/parents>
- <https://www.khanacademy.org/coach-res/parents-and-tutors/student-privacy-for-parent/a/khan-academy-privacy-principles>

We will apply the following specific guidelines as advised by the foundation:

- Moderating youth forum to restrict access to over 18s.
- Data protection for youth user content.
- Privacy of profiles (optional for over 13s).
- Open lines of communication between adults and young people (for example, no private one-to-one messaging).
- Under 13s should not have email addresses and thus must either be initially registered under parent accounts or have the ability to set up an account without email (but this would be difficult to verify and could jeopardise the safety of other under-13 users).

## Development/Deployment

We find that having a local development environment that is easy to set up and configure is crucial to the success of this type of project. It is desirable to minimise any barriers to entry and to make it as simple as possible for developers who want to contribute to an open source project. This aspect of local development is often overlooked, but from our experience it is extremely important that any developer can easily set up a local development environment where they can easily run a local copy of the platform, make changes, and run all unit tests locally.

As we are advocates of continuous integration (CI), we use build servers, typically [Jenkins](#), which build, test and package all commits.

It is also critical to be able to deploy updates to the platform in production in as seamless and pain free a manner as possible. We have found from our own experience that it is also crucial to be able to roll back deploys to a previously known good state.

At nearForm, we use [nscale](#), an open source [Docker](#)-based tool that we developed ourselves to solve both these problems; in other words, to supply the missing piece of the continuous integration puzzle.

## Testing

We are firm believers in Test Driven Development (TDD) and insist on high levels of unit code coverage across both frontend and backend services. We also believe that having good unit tests in place leads to better engagement from the open source community.

nearForm also believe in having good end to end acceptance tests in place from the start, and we like to measure everything:  
[nearform.com/nodecrunch/work-measure-every-week](http://nearform.com/nodecrunch/work-measure-every-week).

Beyond end to end acceptance tests, we also advocate:

- End to end performance tests: recording regular metrics about how responsive the Platform is
- Stress tests: how much load the Platform can handle

We put these best practices in place from the start of development so the metrics are ticking from week 1.

Additionally, we build everything with scale in mind. We have a lot of experience in high scalability, both in terms of collaborating with customers to run 'bees with machine guns' type testing (for which you need special account with permission from Amazon, otherwise you might be seen as a DDOS attacker!), to then analysing the results and making the changes needed to scale out to the appropriate level. While we don't envisage the CoderDojo platform will initially have huge traffic, it will be built with scale in mind nonetheless.

## Design

We have a very experienced and recognised in-house design person, John Rooney (bio below) who is available to assist in all design matters.

We are aware that the new Platform will have the high level CoderDojo nav bar, etc and it will inherit the styles of the main site. We intend to use a CSS toolkit called [Bootstrap](#), which is fast becoming the de-facto tool for modern responsive websites.

## Reporting

The integrations with CRM/Reporting Tool will provide a lot of reports from those Services for the Foundation itself. However, Champions/Mentors will need some basic reports, e.g. basic attendance reports, attendance vs tickets booked, contacts, etc.

Although it is not specifically in the scope of the Requirements, we believe the Platform would benefit tremendously from a very basic reporting engine (even a simple 'search & export results as csv' type solution might suffice in the short term).

## Hosting

We note that linode is referenced in the requirements. While we think linode is fine, a solution based on Amazons AWS might be more applicable in the medium to long term. There are parts of the Platform which may hit performance spikes, e.g. ticketing: consider X thousand tickets being accessed each Saturday or X number of checkins each saturday at a Dojo. This number may be 5,000 this year, 10,000 the following year, etc.

We have a lot of experience in using AWS to scale parts of a system to deal with traffic spikes such as this. It is important to have the correct architecture in place from the start to help deal with such scalability spikes and we find a Microservices architecture to be the most appropriate solution here. This typically involves using auto-scaling groups to horizontally scale certain mircoservices in your system based on certain rules and criteria.

Note that we plan on delivering a cloud agnostic solution (i.e. the platform will not be tied to any underlying cloud, it will run anywhere), however we find the best cloud for this type of platform is AWS.

## Platform Uptime and Monitoring

In order to keep the platform live in production (and to meet a high level SLA), we use best of breed open source tools and techniques for monitoring and alerting of the Platform. We put these tools in place from a very early stage in the development cycle so they become familiar to the client long before the Platform actually goes live.

We also encourage clients to use Performance Management tools if possible, e.g. New Relic or App Dynamics, or we can propose a mixture of open source alternatives if there is no budget for such tools.

Additionally we typically provide 'Run Books' (<http://en.wikipedia.org/wiki/Runbook>) for the whole Platform - these detail all the various services that make up the Platform as a whole from an operational point of view and can be thought of as the operational manual for the Platform. They are an essential part of the handing over of the day to day running of the Platform once the Platform is live and operational.

## How we work

We use an iterative software development strategy. Your system will be delivered in iterations of one week, with a weekly project management meeting. You will also be provided with a weekly demonstration of progress. The production of a weekly live system establishes a natural rhythm that keeps the project on track. It creates a natural focus for activities and keeps everybody honest. It also means that you have a working system at all times that you can use for your own demos.

Our agile process means that you can introduce changes during the project to achieve a better return on investment as you gain more insight into your needs.

However, such changes must be balanced by scope reduction in other areas to ensure the project is kept within the agreed duration. This is explained further in the following article:

<http://www.nearform.com/nodecrunch/how-we-work-the-weekly-project-demo>.

## Open Source

nearForm is open source to our core, as is evident from our GitHub account ([github.com/nearform](https://github.com/nearform)). We are particularly active in the node.js community, both contributing to node.js core, and to many many NPM modules that are actively used every day. Through our involvement in node.js, we have contact with prominent figures in the open source community that we draw on for best practices and also advice (including legal advice). A lot of our lessons in Open Source have been learned the hard way however, and while Open Source is a very simple concept, it's not so simple if not done right from the start. Regardless of who wins the tender, we are happy to advise on the Open Sourcing of the Platform at any point, as members of the CoderDojo community ourselves we'd like to see this done right.

## Community Input

nearForm has a huge amount of experience in building and fostering Open Source communities. We host Node Conf EU every year ([nodeconf.eu](http://nodeconf.eu)) the biggest node.js community event in the world. On a smaller scale, we also run monthly meetups in Dublin for Node ([nodejsdublin.com](http://nodejsdublin.com)), and Microservices ([microservicesdublin.com](http://microservicesdublin.com)), and more recently Node Waterford. We have also run events in the UK with our partner SkillsMatter([skillsmatter.com](http://skillsmatter.com)) including muCon ([skillsmatter.com/conferences/6312-mucon](http://skillsmatter.com/conferences/6312-mucon)).

We have successful Open Source projects of our own ([nscale.nearform.com](http://nscale.nearform.com) & [senecajs.org](http://senecajs.org) in particular) and know what it takes to attract developers to Open Source Projects and building communities around them.

With regards to the CoderDojo platform, we will utilise all our experience in this area to build a vibrant Open Source community around the CoderDojo platform, e.g. we will promote it and talk about it at our own events, and we are also willing to speak about it at CoderDojo events, etc as well as running workshops, case studies, being active on IRC/GitHub, etc. In the long run we plan on keeping up our involvement in the Platform through community contributions ourselves.

# Response to Functional Requirements

The following is our response to sections 3&4 of the Requirements document. As a general principle, we aim to use best of breed open source tools & solutions if they exist and are suitable for each area below.

## **Ticketing**

Ticketing is a major feature of the Platform, and given our struggles with Eventbrite in our local dojo, this is a feature we're looking forward to using ourselves when the Platform is live.

We have looked at several open source ticketing systems and all fail to meet the ticketing requirements of the Platform. To this end, we propose to fully develop the Ticketing system from scratch. We also feel that this will allow us to build a deeper integration with the Platform as a whole, resulting in a much better experience for end users.

We propose to hold a relatively informal workshop with the CoderDojo Foundation and possibly some community members who have experience in this area at the start of the development phase. The result of this workshop will be a set of concrete requirements for the Ticketing system, with MoSCoW priorities ([http://en.wikipedia.org/wiki/MoSCoW\\_method](http://en.wikipedia.org/wiki/MoSCoW_method)) which will be agreed with the Foundation, and we may also take some architectural guidance from community members. While there are a lot of unknowns here we are comfortable with the requirement in general and are willing to be flexible when it comes to implementation.

## **OAuth/Login**

We have used OAuth on several projects, there are best practices within the node community for dealing with OAuth. With regards to integrating with Kata, there is a mediawiki oauth module available and the process for authenticating against a third party service provider seems to be well documented.

Time permitting, we also intend to provide 'log in with Facebook', 'log in with Google', as well as Platform authentication out of the box.

## **Third Party APIs**

We have looked at CRM & Reporting APIs, there are node modules available for all and we do not feel there are any significant challenges in integrating with any of these services. We will work with the Foundation to get all three integrations started as soon as possible in the development cycle.

## **Forums**

We have used open source forum software in the past called NodeBB ([nodebb.org](http://nodebb.org)) and we think it is a suitable tool to act as a foundation for the Platform Forums. It is also written in Node.js and its plugin system allows for easy extensibility.

Additionally it also uses Bootstrap so styling to make it feel part of the bigger site will not be a problem. However, its database is Mongo. While we're big believers in services having their own data stores, we may look to implement a PostgreSQL database for it.

## Mozilla Open Badges

This is new to us. We have studied the Open Badges API and are reasonably comfortable with it. However, it is not entirely clear how it will work in the Platform, i.e. what exactly the user flows are, etc. We propose doing a workshop around this at the start of development also, and again, the output of that would be a concrete set of requirements with agreed MoSCoW priorities.

## Zen Reproduction

We don't foresee any major challenges with the Zen Reproduction phase, it should be very straightforward. The database migration script, front end work, and back end service implementation can all happen in parallel. From a development perspective, we intend to get Zen Reproduction complete end-to-end in the first few iterations. This will create a solid foundation upon which to build the rest of the platform.

## Misc

We are comfortable with the rest of the requirements, i.e. the Champion onboarding/profile creation/Champion registration, etc. Likewise Dojo Listing, Mentor Profiles, Dojo Admin, etc will be very straightforward development wise. All of these will be modules/plugins based, and they will be part of the core plugins that all Dojos get. For the Dojo Mail, we have experience with the Google Mail API and are comfortable with that also.

## Comments

From our personal experience in the local CoderDojo, we find there is a lot of 'double entry' of data that we need to do to keep track of kids, multiple parents contact details, consent forms, attendance details, and a whole lot more. We think the proposed new Platform will really help with this, and we will be 'dogfooding' the new Platform every step of the way with our fellow local Mentors and Champions. Locally, we also get feedback from the Mentors that they would like recognition on LinkedIn that they are involved in CoderDojo. We are hoping that the Mozilla Badges can be applicable for Mentors also, and that they can be displayed on LinkedIn.

# nearForm Profile

nearForm was established as a company in November 2011 by “Mobile Apps in the Cloud” author Richard Rodger (former CTO for FeedHenry) and Cian Ó Maidín. The company employs staff in Ireland, the UK, Europe and the US, their main office is in Tramore, Co. Waterford, Ireland. The company is heavily involved in the node.js community and are dedicated to the evangelism of the node.js system and its adoption by the wider software community.

The team at nearForm has delivered over 50 production Node.js systems to many market leaders spanning Media, Banking, Enterprise, Travel, and many more.

nearForm have been running Node.js in production and at scale since Node v0.4. It is estimated that code written by nearForm staff is used in 40% of production Node.js systems around the world.

Company name and address:

nearForm limited,  
Tankfield,  
Tramore,  
Co. Waterford.

As a company the nearForm pedigree is not that of a web company turned consultancy. The nearForm team are an enterprise-strength, battle-hardened product development group who come from core systems development for enterprises ranging from statistical analysis software, enterprise mobility, telco virtualization, banking, enterprise app development, consumer-facing web services, and a range of other technology stacks. The nearForm ethos is to work only on projects where the team adds significant value and can have a large impact. It is also to build the most user friendly and usable technology solutions for human use. The team is of exceptional quality and provides full transparency in every engagement.

## Statement of Capabilities

We believe that our combination of a strong team, the best practices, CoderDojo community involvement, proven methodology and an excellent track record of delivery makes us an ideal partner to develop the new CoderDojo platform.

### **Proven Track record of delivery**

Our company, nearForm, as its main business activity, builds complex systems. We do this for start-ups and for established companies. Sometimes, delivering even a small feature set means conquering a mountain of hidden technical and structural complexity. We have delivered more than 50 production systems in just under three years. Our customers know they can rely on us to deliver in complex environments, with many stakeholders, on time, on budget and with a high standard of quality.

### **Significant competence in the area of Platform Development**

We believe that because we go to great lengths to find the right expertise that we can deliver projects at a higher quality, faster and with less risk than others. The nearForm team has grown to nearly 40 employees and currently boasts three published authors. We have built a strong competence in large scale systems that involve Web Service Platform Development and integration.

### **Client centric approach to delivery**

We understand the conflicting dynamic many large companies face when it comes to implementing a large new feature set. You must plan and budget in advance to ensure you get what you require within an acceptable budget, however in software

it is impossible to answer all questions or indeed finalise all requirements before a project starts. Our process keeps you at the locus of control so that at every step throughout the project you can guide the team to ensure that the results exactly match your needs and expectations.

## Implementation Plan

At nearForm we make use of leading design & development technologies. We follow the agile development approach for design quality, flexibility and ease of development & testing, beginning with a requirements analysis, project design and planning, implementation, quality assurance through strict testing, and finally deployment. nearForm will take the project from analysis and design through to deployment, ensuring effective planning, team communications and feature delivery throughout each stage of the project.

Weekly product demonstrations mean you will be able to guide the project as new information and feedback arrive.

Daily scrum meetings allow nearForm teams to keep up to date on the project, members of the Foundation are also welcome to join if they so wish (all our work is done in an open and transparent manner). A weekly meeting will take place outlining what features have been completed. This will ensure good communication between both teams resulting in sprints being delivered as described in the project plan.

### High Level Project Schedule

As per the Requirements, the Project will be available for beta use in June 2015, with final delivery in September 2015.