**Abstract**

Electron is an open-source framework developed and maintained by GitHub. Electron allows for the development of desktop GUI applications using web technologies: It combines the Chromium rendering engine and the Node.js runtime. Electron is the main GUI framework behind several notable open-source projects including Atom, GitHub Desktop, Light Table, Visual Studio Code, and WordPress Desktop. The fact that the project is under active development can be deduced from the speed at which pull requests are merged. Until 2019-10, it has 22559 commits, 140 branches, 628 releases ,1036 issues, and 889 contributors.

**Introduction**

Electron allows for the development of desktop [GUI](https://en.wikipedia.org/wiki/GUI) applications using web technologies, such as javascript , html and css. This chapter gives a structured overview of Electron by providing descriptions of the various parts that compose its architecture, in order to explain the system and how it works. Several aspects of electron are analyzed, starting with the stakeholders. After that We give a context view to describe the different components involved and a development view to outline how electron is developed and the ways in which contributions can be made.

**StakeHolder**

Many stakeholders played a part in the project Electron.

**Assessors**

Assessors oversee the system's conformance to standards and legal regulations. And we can also say assessors are the core developers. The team of this project is consisted of 6 core developers and many other developers. As the projects are still in development Control the quality of submitted code and guide the project to the right direction step by step.

**Communicators**

The people or introductions that explain the architecture of the project are the communicators. The biggest communicator is the electron development team , called Work group(WG). Now there are 6 core members. And chairman is [@jkleinsc](https://github.com/jkleinsc) . Besides, the community—github plays a very important role, who provides the place where developers can put up the documentation and get the bug reports about the project through the issue tracker.

**Developers**

There are two kinds of developers. The one is core developers, the other is contributors.

**core developers:** they hand up most of the commits, The success of the project today can not be separated from their efforts. @zcbenz, who have handed up 6086 commits , and @kevinsawicki , who has handed up 2138 commits, are the core developers of electron. They contributed the most to the project before 2017 and after 2017, respectively.

**contributors:** they contributed the development of the project more or less. And their commits need to be examined by WG. Until 2019/10/3 , there are 889 contributors.

**Maintainers**

The maintainers of the project are @jkleinsc and his team will deal with all the problems and bugs after checking the bug reports, oversees all release branches, and tooling to support releases.

**Support Staff**

Also the WG, they need to:

Releasing Electron according to schedule

Release timeline coordination

* + When to cut new branches for major and minor release lines (e.g. v3.0 -> v3.1)
  + Planning beta cycles, timelines, etc.
  + Listening and responding to feedback

Management of Heroku apps for associated repositories.

**Users**

There are mainly two kinds of users:

**Workers:** This project can be used by developers to develop many desktop projects.

**Learners:** In the term of education, it can be used both in teaching and learning, such as javascript, html , css and so on.

**Sponsors**

Electron used to be called [Atoms shell], it is used to develop the editor

[Atoms] by github , so github is the inventor ,and also the biggest sponsor of electron. Besides, many individual developers sponsored the project. They definitely support the project and their or their companies’ logo will be shown in the Github.

**Competitors**

Other frameworks for desktop application development, like QT, etc.

**Context view**

The diagram below in Figure 1 shows the context view of Electron. The view shows the different relations, dependencies and interactions Electron has with its environment. In this section we will provide some short explanations.



*Figure 1: Electron context view*

**Development tools**

You can use the native Node. JS development environment to develop Electron applications. In order to build an Electron desktop development environment, you only need to install Node.js, NPM, a handy code editor, and a basic understanding of your OS command line client.

We recommend that you use GitHub's Atom or Visual Studio Code. These two popular editors are developed by Electron. And support for JavaScript is great.

**Test**

Electron is based on Chromium, so you need a display driver to make it work. If Chromium fails to find a display driver, Electron will fail to start, so no matter how you run it, Electron will not perform any of your tests. When testing Electron-based applications on Travis, Circle, Jenkins or similar systems, some configuration is required. Essentially, we need to use a virtual display driver.

**Communication tools**

Electron uses Github and Git for version control to help developers collaborate and track issues related to the software. Contributions in the repository come from the Electron software development team, as well as from the open-source community. In order to stay in contact with developers and users the community of Electron are built.

**Software developed by Electron**

If you can build a website, you can build a desktop application. Electron is a framework for creating native programs using Web technologies such as JavaScript, HTML and CSS. It is responsible for the more difficult parts. You just need to focus on the core of your application. With Electron, you can easily create desktop applications or ideas for your company.