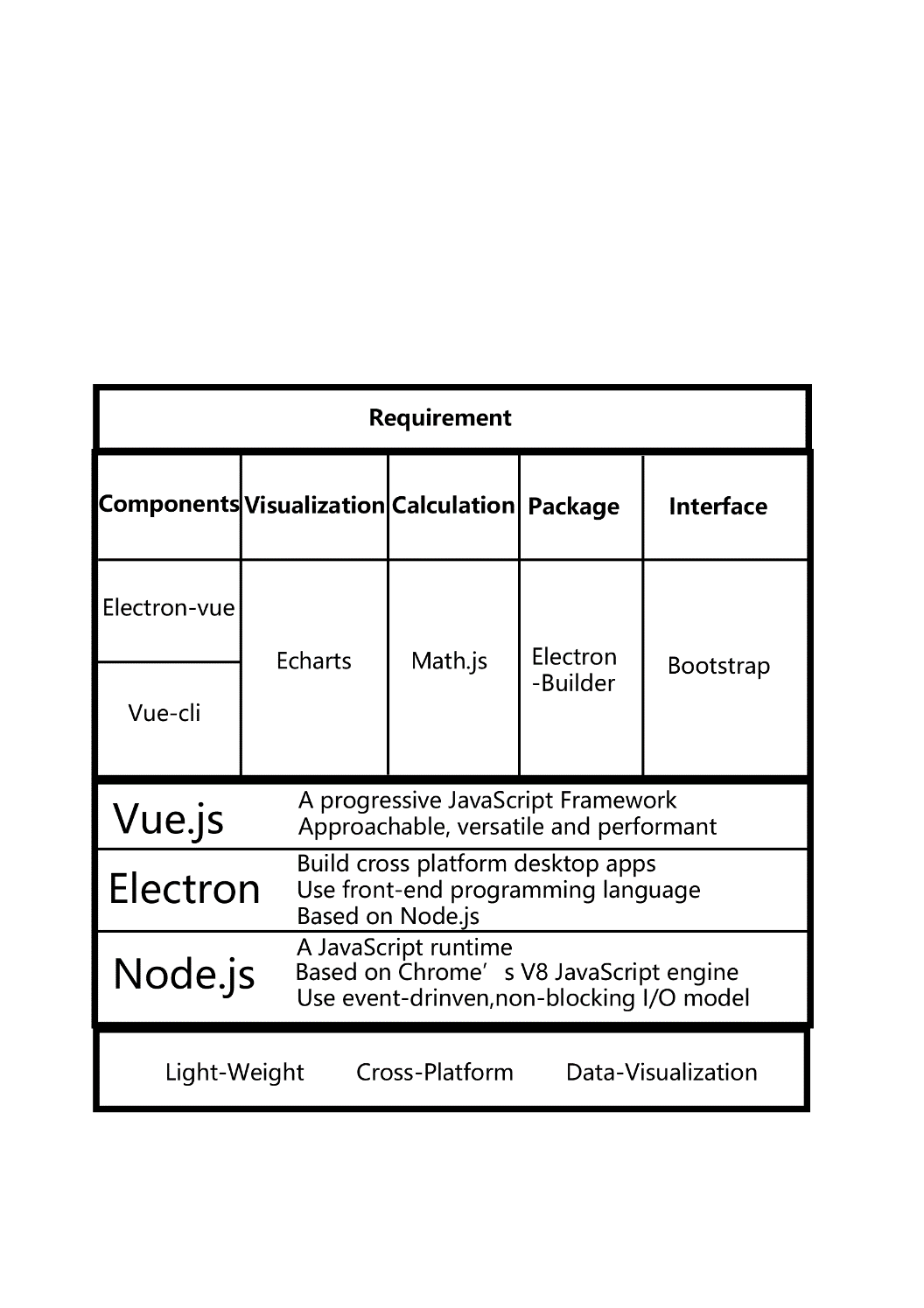
**Introduction**

Considering the requirements and developing cost of the whole project, we chose Node.js and Electron as develop platform and Vue.js as front-end framework. The coop-work of those techniques help developing a light-weight and cross-platform app which works as a algorithm simulator and data visualization tool.

Other tools, such as Echarts.js, are used for satisfying specific requirement as below image shows technique selection structure.



**Develop Language**

JavaScript, HTML and CSS

As this project requires data visualization function, front-end is a suitable choice as there exists visualizing frameworks like Echarts.js based on JavaScript and canvas in front-end development.

Front-end developing also decrease the cost of interface development and can easily improve UI by CSS tools, such as Bootstrap.

**Develop Framework**

Electron:

Electron is an open-source framework to build cross-platform desktop Apps with front-end language. It is based on Node.js and npm source.

Official site: https://electronjs.org

Vue.js:

Vue.js is a progressive JavaScript framework. It is approachable, versatile and performant.

The key advantage is make the software into components, which is much easier for group project as group member will not get confused by reading others’ code – they just work on their own component.

Official site: https://vuejs.org/index.html

**Develop Tools**

Vue-cli:

Vue-cli is a full system for rapid Vue.js development and aims to be the standard tooling baseline for the Vue ecosystem.

By using this tool, the components structure can be much clear and easy to coop-develop.

Official site: https://github.com/vuejs/vue-cli

Electron-Vue:

The aim is to remove the need of manually setting up electron apps using vue-cli.

Official site: https://github.com/SimulatedGREG/electron-vue

**Package Tool**

Electron-builder:

A complete solution to package and build a ready for distribution Electron app with “auto update” support out of the box based on npm/yarn source.

By using this tool, we can easily package the software into cross platform installation package and installation-free software. The size of package will also be small. In addition, auto-update function is also supported by this tool.

Official site: https://www.electron.build/

**Data Visualization Tool**

Echarts.js:

ECharts.js is a powerful, interactive charting and visualization library for browser.

The API of this framework can satisfy all requirement of the software.

Official site: http://echarts.baidu.com/

**Math Tool**

Math.js:

Math.js is an extensive math library for JavaScript and Node.js and is used in this project to do matrix calculation.

Official site: http://mathjs.org/

**Interface Tool**

Bootstrap:

Bootstrap is a popular front-end component library to help complete the interface of the software.

Official site: http://getbootstrap.com/