

Theme: Next Generation Web

Subject: Ultra-lightweight Web Application Model

Introduction

Traditional web application model is designed for PC and Mobile devices that usually have much greater computing power and h/w resources than ultra low-end devices (such as wearable devices). The goal of this research project is to define and develop an ultra-lightweight web application model that will allow compliant web applications to run on very limited computing power and resources. The ultra-lightweight web application model should be backward compatible with standard web application model, which means that the ultra-lightweight web applications should be able to run on normal Browsers.

The ultra low-end devices may or may not have display devices attached with them. The RAM size of an ultra low-end device can be either 64MB or 128MB. For 64MB device, the available system memory (after device boot-up) is expected to be 10~15MB, while for 128MB device the available system memory is expected to be 35~45MB. The typical CPU power for an ultra low-end device is 200MHz (single-core or dual-core).

The “ultra-lightweight web application model” should be designed for the above-mentioned ultra low-end device segment.

Scope

Challenges that define and develop an ultra-lightweight web application model for the fast growing segment of ultra low-end devices include:

- Definition and specification of ultra-lightweight web application model
- Method to optimize existing open source software or design a new software to support the model;
- Verification of working model;

Research questions

We are interested in the following research questions. These questions are not exhaustive but different research questions are open to discuss with research partners.

- What would be the most essential Web specifications/features that we have to keep in this model for developers?
- What would be the most effective way to control the memory usage Web engine considering the dynamic nature of DOM tree and other internal data structures?
- What would be the most effective way to control the memory usage of JavaScript engine considering the dynamic nature of JavaScript language (such as prototype chain)?
- What would be the most efficient method for Garbage Collection in this model?

Expected Deliverables

The following is open to discussion:

- Specification of the ultra-lightweight web application model;

- Detailed progress reports every 3 months summarizing accomplishments.
- Prototype implementation
- Patents with Samsung Electronics (if agreed)