# Gen 2
# Technology & Systems Engineering
## plus platform functionality

# what we've evaluated & delivered

MUST-HAVE TECHNOLOGY COMPONENTS FOR INTEGRATION

ROLE OF WIFI, BLUETOOTH, AND SENSORS - AND WHAT THEY DO

PRICING OF COMPONENTS PLUS PROTOTYPING/PROGRAMMING

THE CONSUMER-FACING PLATFORM / FEATURES  FUNCTIONALITY

REQUIREMENTS FOR BUILDING A 1.0, 2.0, 3.0 VERSIONS

RANGE OF MAGNITUDE / TIMING

# Prevailing Technology Strategy

Repurpose existing technology.

Simple on the outside, smart on the inside

Resist the “Swiss Army Knife”

Intelligent. Useful. Intuitive.

Phased functionality

Informed by what Consumers told us.

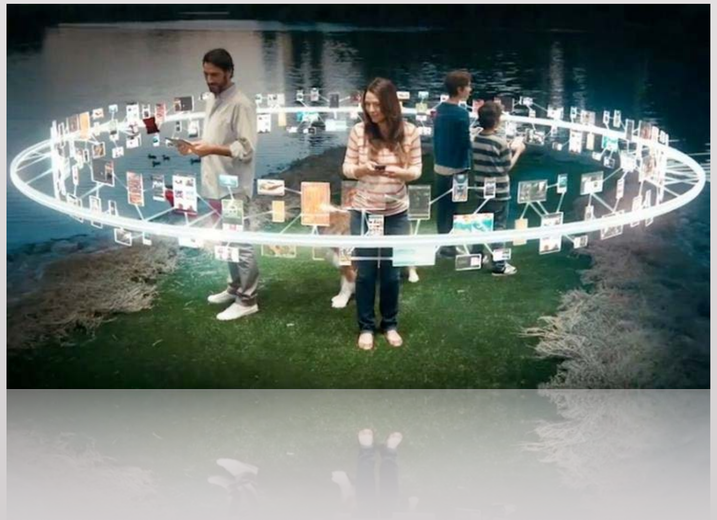
# Technology & functionality designed for Health & Wellness

## Sensors.

Readily available, extremely small and cost effective

Sensors deliver an intelligence through proximity, weight, flow, temperature, tilt, free-fall (gravity), motion, pressure.

**User experience:**
**Intelligent monitor of system, usage to inform consumption, reordering, modes, etc.**



## Connectivity.

WiFi is now allowing appliances to enable connections that mesh within a home or office giving way to smarter, appliance-based networks.

Wireless dialog so that devices can talk to one another. Low energy bluetooth triggers alerts, and serves as a back up should WIFI .

**User Experience:**
**Contextually relevant message, non intrusive, device neutrality**


## Mobile interface.

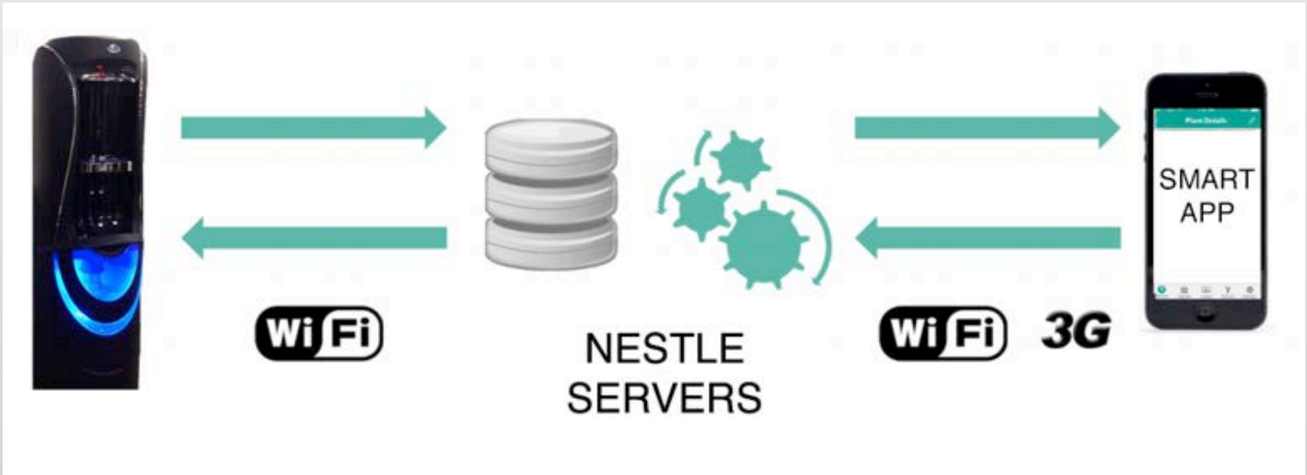
The mobile device delivers the interface, versus the Water Dispenser itself

This will then become the “remote control” of the system and will report usage

**User Experience:**
**Advanced system control through mobile/ internet**

# 1. Wifi + Blue Tooth = Best Practice






| Blue Tooth | Blue Tooth + WIFI |
|---|---|
| Allows dispenser to communicate in proximity with user via mobile device | Allows the unit to communicate directly with Server for range of diagnostics |
| Requires an “App” to be running to receive alerts/messages | Upload usage stats so that user can view them on line, or within mobile interface [“app”] |
| Requires users to be within a | Requires a WIFI network [not a usually a problem in homes] |
| Limited to simple messaging functionality that is less intelligent than what is capable with WIFI | Blue tooth and Mobile Interface together configure to local network, to set up the connection |
| Can be used in combination with WIFI to configure the dispenser or for user authentication | Can fall back to Blue Tooth to continue to report usage stats in the event WIFI connectivity fails or is out of service |
| The low-power of Blue Tooth alone is not an advantage given the Dispenser runs on power | unlimited range between Dispenser and User - as long as they’re both connected to internet - alerts and status can continue |
| | Server is the Hub which connects the Dispenser and the Device<br>Nest and Wink are best practice use cases |

# 2. Technology components

“CPU”/micro processor [Spark Core]
- ☆ intelligence of the system
- ☆ board that brings the components together

Blue tooth and Wifi
- ☆ activates the app, proximity sensors, and connects to the server to send / receive data

Flow Sensor
- ☆ measures frequency of pour / dispensing

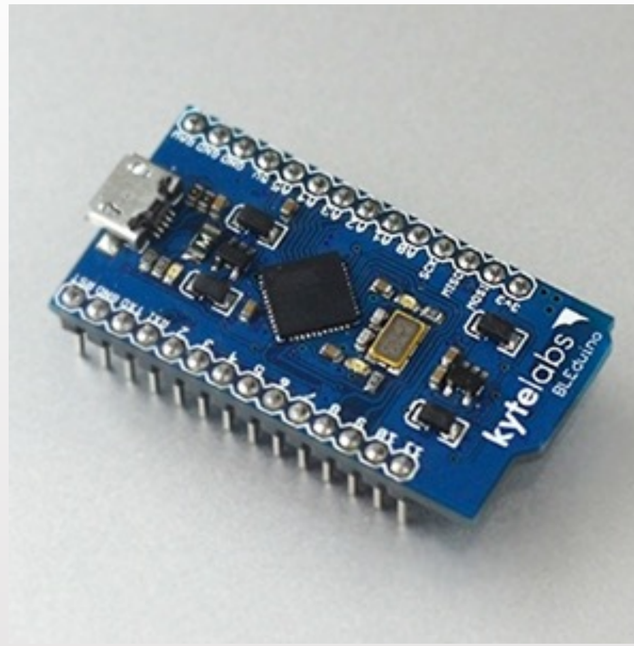
Level Sensor
- ☆ measures quantity consumed by weight of water under the bottle

**Note:**

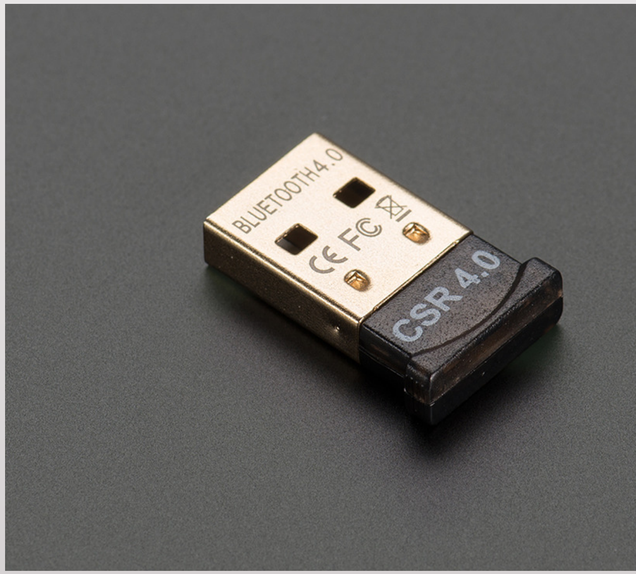
We may be able program the existing sensor technology embedded in the prototype, for dual purpose; further investigation can confirm.

spark


bleduino-kytelab




G14Wate Sensor

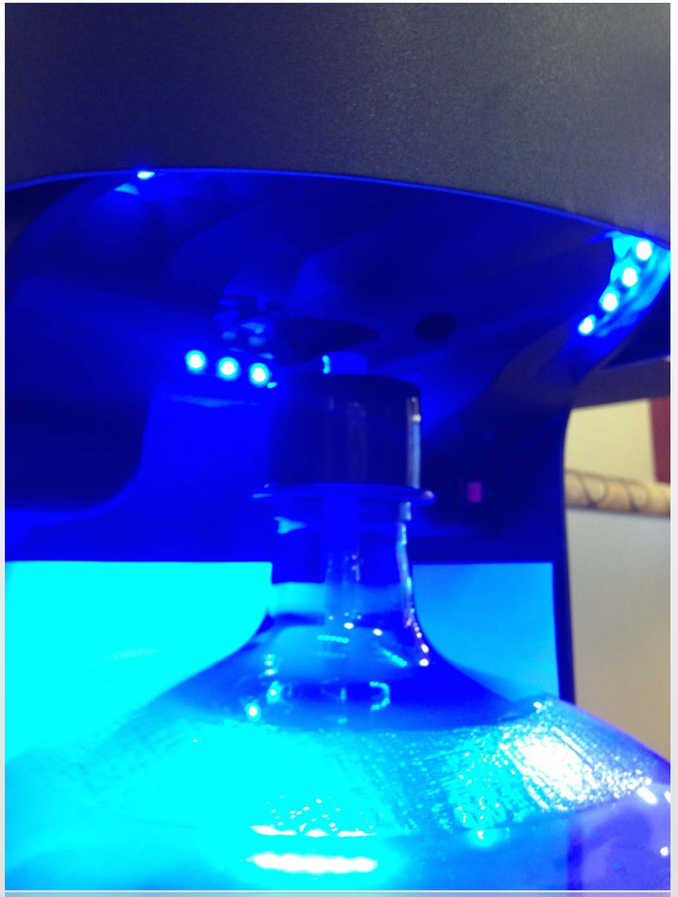



Bluetooth BLE

# 3. Integration in planned unit

Exact placement, to be determined - but inside and back is most probable for integration in the existing unit:

- Sensor for flow needs to go inline with water feed; connects to feed hose
- Sensors can measure level of water by weight so bottom placement would be required
- Place sensor technology along side, or with sensors already installed - if possible



# Hardware Cost per Piece; Programming / Prototyping

| INDIVIDUAL COMPONENTS | Priori | Supporting Feature(s) | QTY | Prototype | 10k Unit PO | 40k Unit PO | CORE FUNCTIONALITY |
|---|---|---|---|---|---|---|---|
| ARM (SparkCore) | yes | Brains and WIFI Connectivity | 1.00 | $50 | $4 | $3 | Backbone to the functionality and serves as the Operating System for unit; can receive electronic (digital or analog) signals |
| CC3000 WIFI Module (TI) | yes | Brains and WIFI Connectivity | 1.00 | $45 | $17 | $15 | delivers intelligence and connectivity on line; allows mobile integration |
| Bluetooth SMD Module - RN-42 (v6.15) WRL-12574 | yes | Bluetooth Low energy | 1.00 | $50 | $12 | $9 | complements the wifi; sensors "run" with blue tooth |
| Water Flow Sensor G1/8" TEM01072B | yes | Measures flow of water | 1.00 | $20 | $9 | $7 | Sensors that support certain feature types; recongition, detection, auto away; does it have a personality |
| Water Level Sensor | yes | Ties into existing sensor or senses on its own | 1.00 | $20 | $14 | $11 | Variable depending upon how the unit is designed. Chance we may be able to integrate with current sensors already planned in prototype |
| Proximity Sensor IR | yes | Detect human presence | 1.00 | $25 | $6 | $5 | Activatesmobile app or initialize ready status. Can also initiate auto-away status. |
| PC Board, micro connectors / cablets | yes | "Houses" all components | 1.00 | $50 | $31 | $25 | Component integration |
| Total (Hardware) | | | | $260 | $93 | $75 | |

| REQUIRED LABOR/MANHOURS | DELIVERABLES | ROM |
|---|---|---|
| Prototyping | Assembly / Fabrication of Protoype | $5K - $7.5K |
| Programming of components | Programming logic / triggers | $20K - $25K |
| | States diagrams | |
| | Programming software intermediary | |
| | Enable connectivity to server | |
| | Writing code that enable server data to connect to mobile | |
| QA Testing | Prototype, Software AND together | $7.5K-$10K |
| | Subtotal | $32K - $42.5K |
| *QA Allowance* | *Engineering tweaks required to retool kinks or deficiencies* | *advise $10K* |

# HOW WE MIGHT BUILD THE CONSUMER [USER] FACING PLATFORM AND EXPERIENCE

NOW V1.0 / NEAR V2.0 / NEXT V3.0

# High level Functionality “Buckets”

| 1.0<br>Basic Consumption<br>[report-remind] | 2.0<br>Personalization<br>[sense-interpret] | 3.0<br>System Intelligence<br>[sense-interpret-manage] |
|---|---|---|
| Intelligence [rasperri pi or spark + wifi]; Bluetooth module | Intelligence as in 1.0; Bluetooth module. Phone control surface for info relay. | Intelligence as in 2.0; Bluetooth module. Mobile app for notification, interpretation and management of health/wellness. |
| Aggregated consumption - reported against a set of criteria | Individualized consumption based on unique cell phone number [or UDID] | Tie into Nestle customer database. |
| Local alerts to consumer for cooler maintenance. Same as LEDs on cooler. Change the water, change the filter… | Alerts to individuals health and wellness. “It’s time to drink.” | Track bottle changes and bottles (full vs empty) and notify customer if they need more. |
| Emailed usage reports. a la Nest | Emailed usage reports. a la Nest | Plug it into online reordering process. |
| Reminders to consumer to drink more water [IR sensor] | Prompts for ordering based on consumption trending | App info and website share and replicate information. |
| Allows Nestle to track household consumption data: when, quantity | | Track and monitor usage based on time and day. Plan wellness and compare to averages. Share. |

# Version 1.0 - Basic Consumption [report/remind]

**Technical / build requirements** : iOS, Android (TBD).

App is a browser interface that will assign an IP address to the Dispenser

Will access already existing Nestle web information (customer info)

**Data Integration -** app will access existing information (continuity from web page to phone). Settings page to configure water cooler to local network.

Data integration and calls to server done through REST calls rather than through an app.

**How would this be managed?**

IP address assigned by cooler technician. App downloaded via app store. Data information pulled from Nestle site.

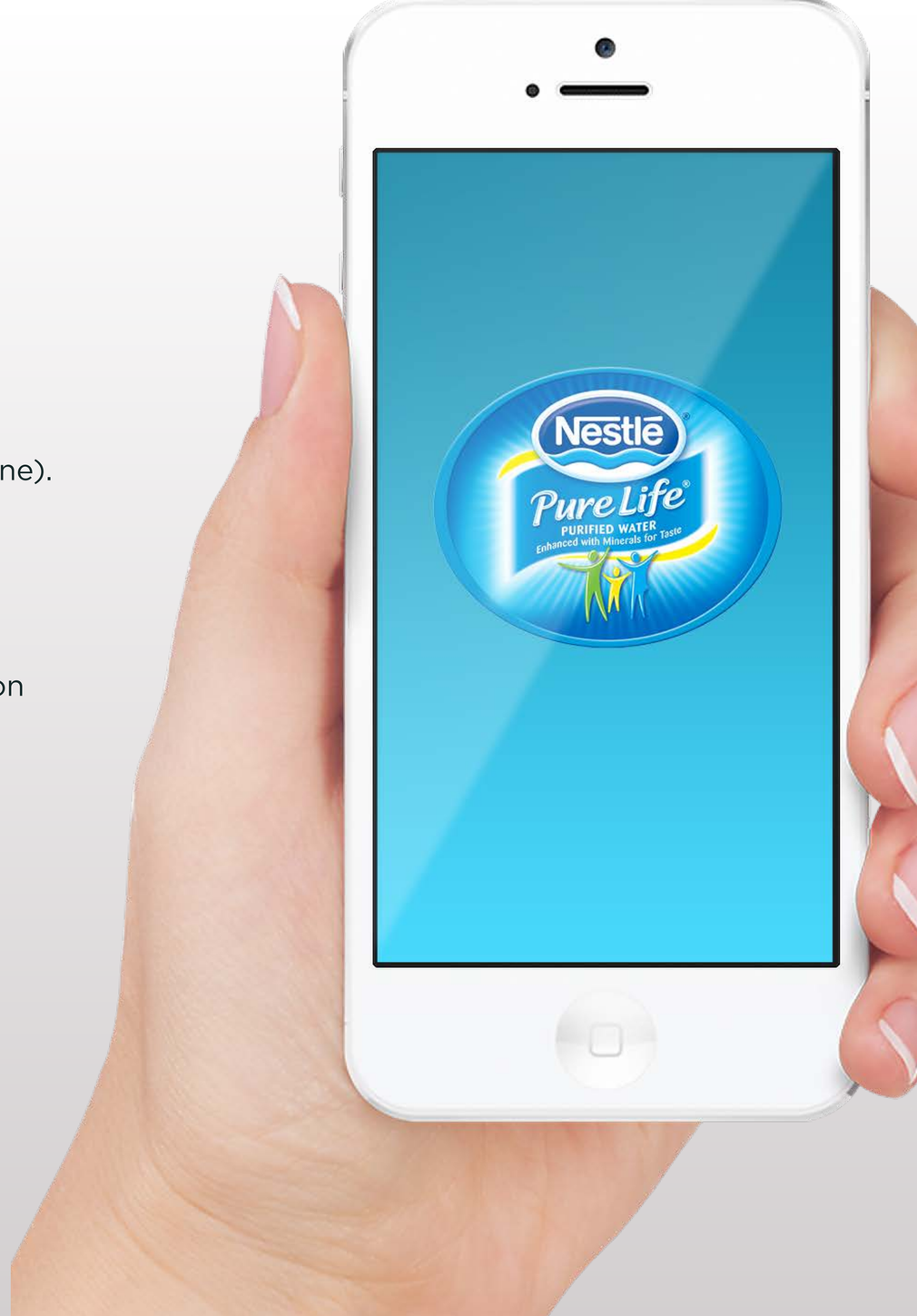
**Timing:**

**8-10 weeks:  rough prototype Beta**

**4-6 weeks: remaining development**

**1-2 weeks: App store submission (as required)**

**ROM cost for a basic platform “app let”? $75K - $95K (TBD)**



# Version 2.0 - Personalization [sense/interpret]

**Technical / build requirements** : iOS, Android (TBD).

App is a browser interface that will assign an IP address to the cooler

Will access already existing Nestle web information (customer info)

App can sense and interpret data (managed/editorial by Nestle)

**Data Integration** -

Reports aggregate usage per household.

Reports individual usages by user (manually added by account holder)

Notification of usage

Tips/hints from Nestle data (opt in/out)

Event based notification (heat wave, Olympics, marathon)

**How would this be managed?**

Dashboard and email notifications of usage

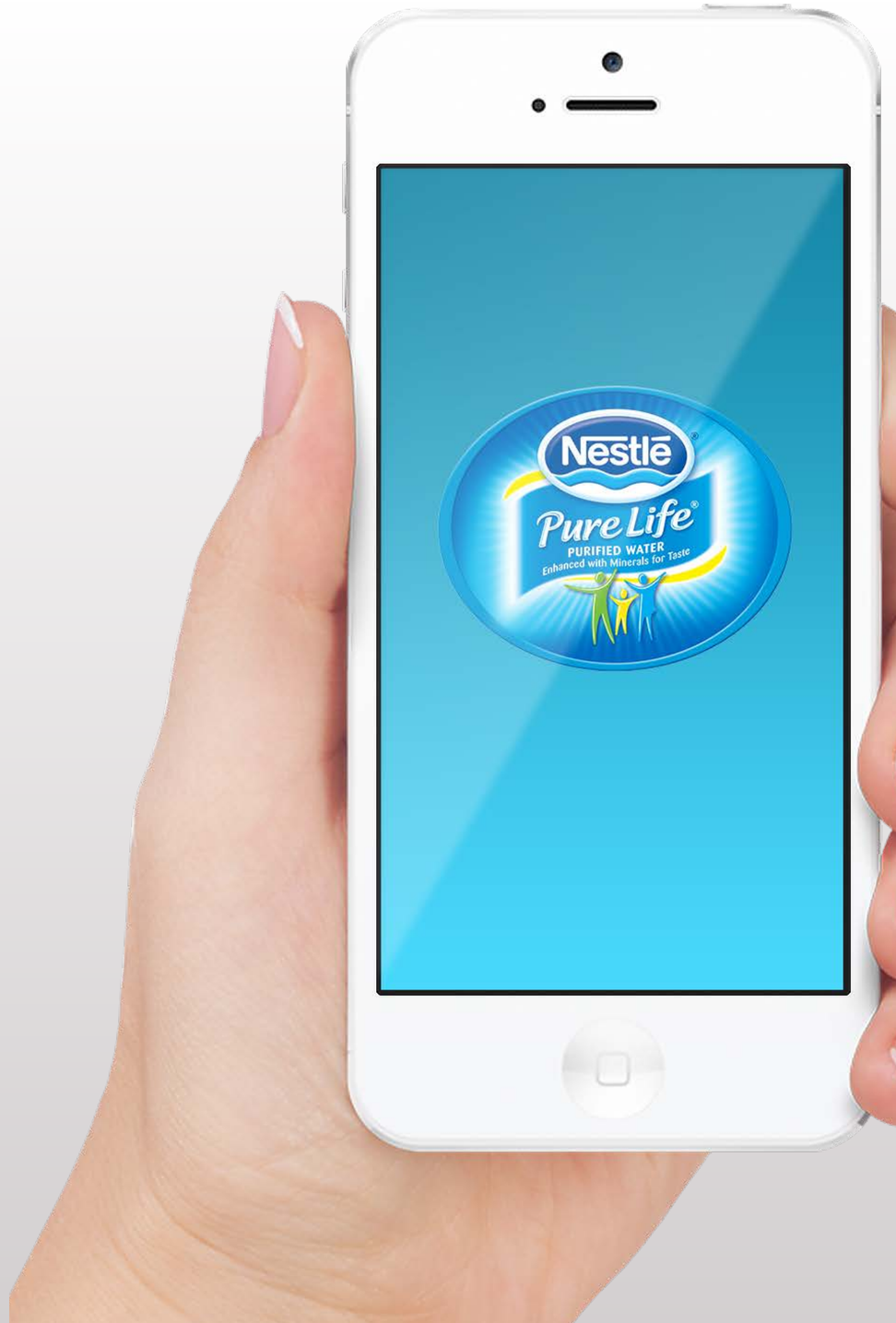
**Timing:**

8-10 weeks:  rough prototype Beta

4-6 weeks: remaining development

1-2 weeks: App store submission (as required)

ROM cost for a basic platform “app let”? $100K - $125K (TBD)



# Version 3.0 - System Intelligence [sense/report/manage]

**Technical / build requirements** : iOS, Android (TBD).

App is a browser interface that will assign an IP address to the cooler

Will access already existing Nestle web information (customer info)

App can sense and interpret data (managed/editorial by Nestle)

**Data Integration** -

Reports aggregate usage per household. and individual usage (manually added by account holder)

Notification of usage to prompt ordering

Tips/hints from Nestle data (opt in/out)

Event based notification (heat wave, Olympics, marathon)

**How would this be managed?**

Dashboard and email notifications of usage

Account managed by user (wellness planning, training schedules, etc)

Achievements and badging (dashboard)

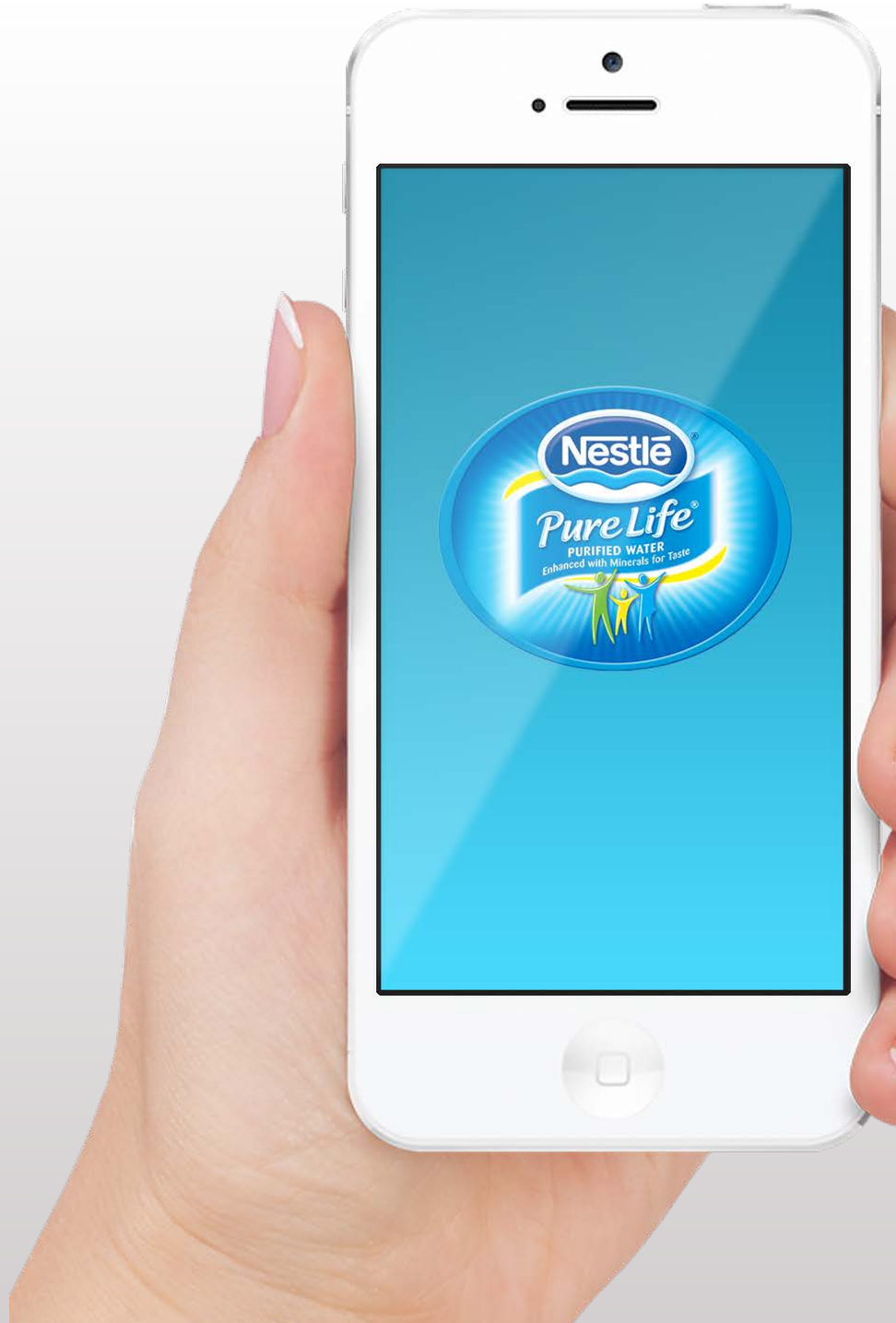
**Timing:**

16-18 weeks:  rough prototype Beta

6-9 weeks: remaining development

2-3 weeks: App store submission (as required)

ROM cost for a basic platform “app let”? $150K - $175K (TBD)



# Use Case Example: Nest types of reporting possible!

nest

## December Energy Report

nathan@urbanvisuals.com

Since October 2011, all Nest Thermostat schedules have saved:

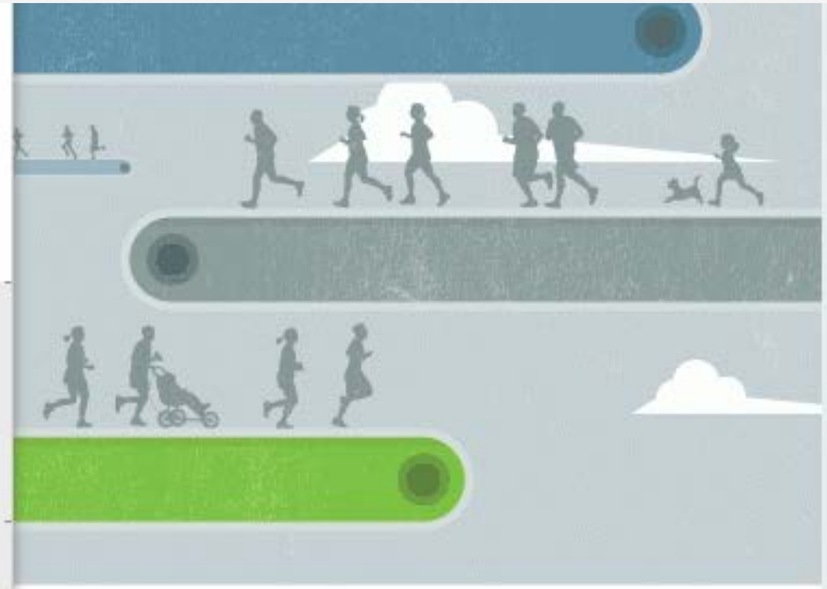
**3,446,518,428** kWh

That's the same amount of energy it would take for everyone in the US to exercise hard for 91 hours straight.

Energy savings are an estimate, not a guarantee that you will save energy. Learn more >

Share how much energy Nesters are saving together.

Share Tweet



Welcome to your Nest Energy Report. It tells you how much energy you've used to heat or cool your home and has tips to help you save more. Learn more >

We're looking at available data from: Main.

## Here's how you did:

This month you used 6 fewer hours than last month. **-6** hrs

November **86** hrs

December **80** hrs

Tell your friends how much energy you're saving. Share Tweet

## Why did your energy use change?

We look at a lot of reasons your energy use can change — from weather to Auto-Away — and these are the ones that made the biggest difference this month.

They add up to -7 hours of energy use. The difference of +1 hour was caused by other factors. Learn more >

| -8 hrs | +3 hrs | -2 hrs |
|---|---|---|
| You set your Nest Thermostat to Away more often this month. | December had more days than November. | You adjusted the temperature to use less energy this month. |

## A tip for you:

Want more Leafs?