Improving **Energy** Efficiency

Peter Gammelgaard

Developer at **Shape**
@PeterGam

Why?

Your Obligation as a Developer!

– Apple Documentation


Low Battery
10% of battery remaining
Dismiss


Settings
Battery
Low Power Mode temporarily reduces power consumption until you can fully charge your iPhone. When this is on, mail fetch, Hey Siri, background app refresh, automatic downloads and some visual effects are reduced or turned off.
Battery Percentage
Show percentage of battery remaining in the status bar.
BATTERY USAGE
Last 24 Hours
Last 7 Days
Safari 19 %
Beam
Background Activity
16 %
Photos 8 %
Tweetbot
Background Activity
6 %
Seven 5 %
App Store
Background Activity
5 %

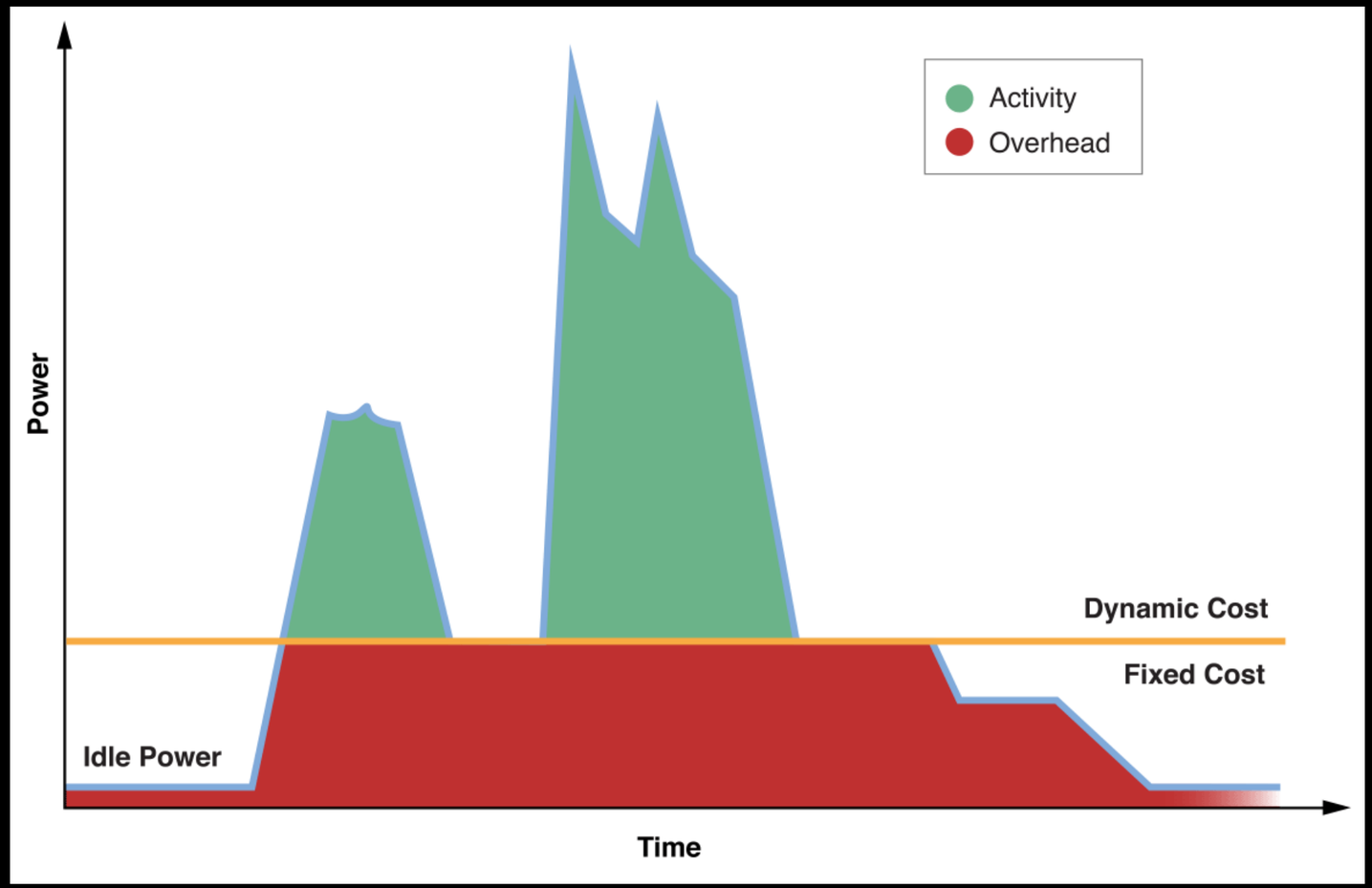
Motivation

- We had an app that *consumed* way to much **energy**
- Showed up on top inside **Battery usage**
- Device got warm when using the app for a while

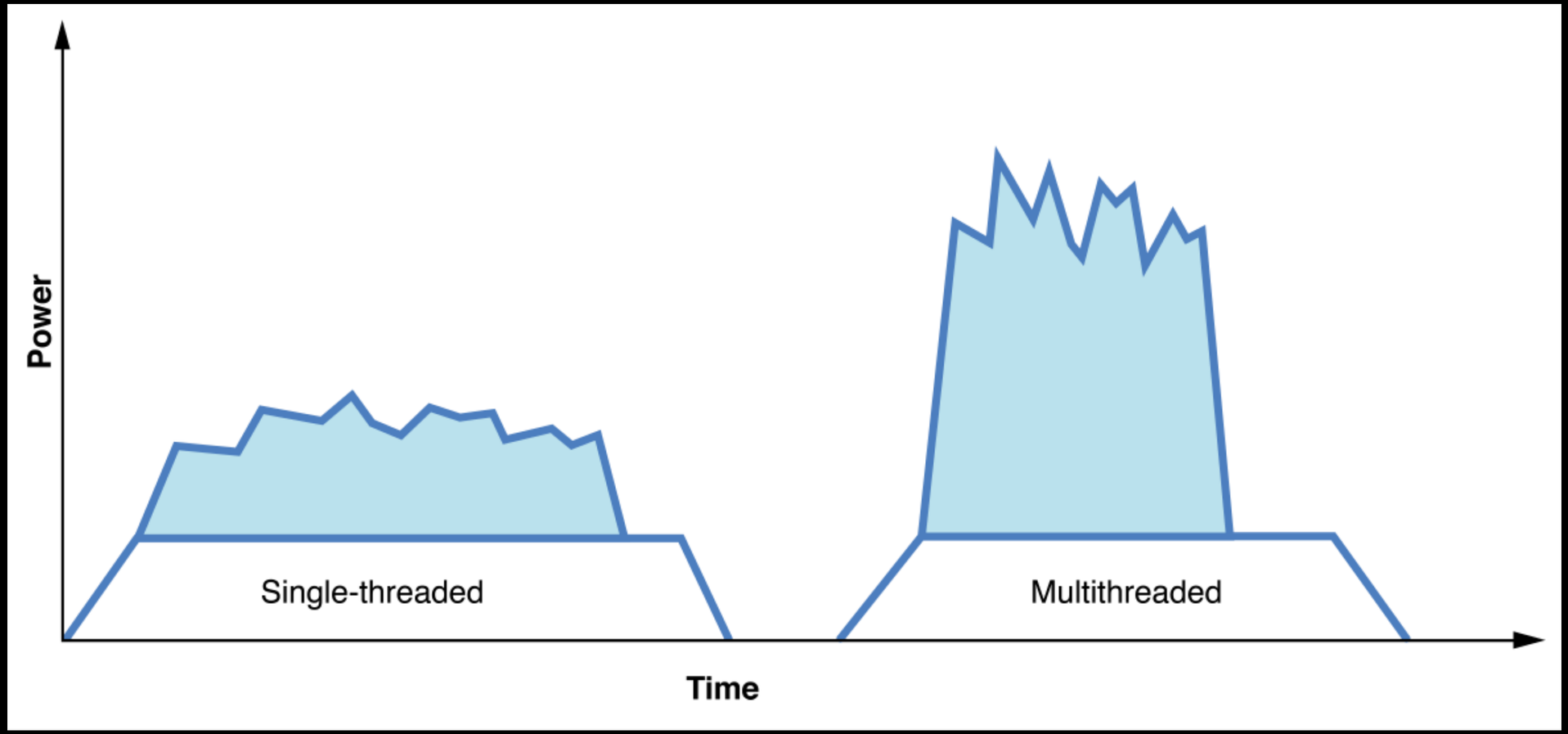
Fundamentals

- There no single solution
- Dependent on the features of the app
- Bug? Feature?
- Best practices

Fundamentals



Fundamentals



The app

- UIKit based application
- Lots of JSON data feeds that expire from cache often with different interval (15 seconds, 30 seconds, 1 min, 5 min, 10 min, 30 min, 24 hours)
- Visible data must be kept up to date without user interaction (no pull to refresh!)
- Bunch of data feeds visible at once

Challenges

- Many network requests
- Constantly polling for several data feeds
- Parsing a lot of data

Goals

» Improve **battery** usage until **acceptable**

» **Transparent** for **user experience**

» **Low-hanging fruits** approach

Goals

- Improve **battery** usage until **acceptable**
- **Transparent** for **user experience**
- **Low-hanging fruits** approach
  - Rewrite of app not feasible
  - Rewrite of backend not feasible

To the documentation!

What uses energy?

- CPU
- Animations and drawing
- Networking
- Background usage
- Location
- Motion
- Bluetooth

What uses energy?

- CPU
- Animations and drawing
- Networking
- ~~Background usage~~ 💪🏽
- ~~Location~~ 💪🏽
- ~~Motion~~ 💪🏽
- ~~Bluetooth~~ 💪🏽

CPU

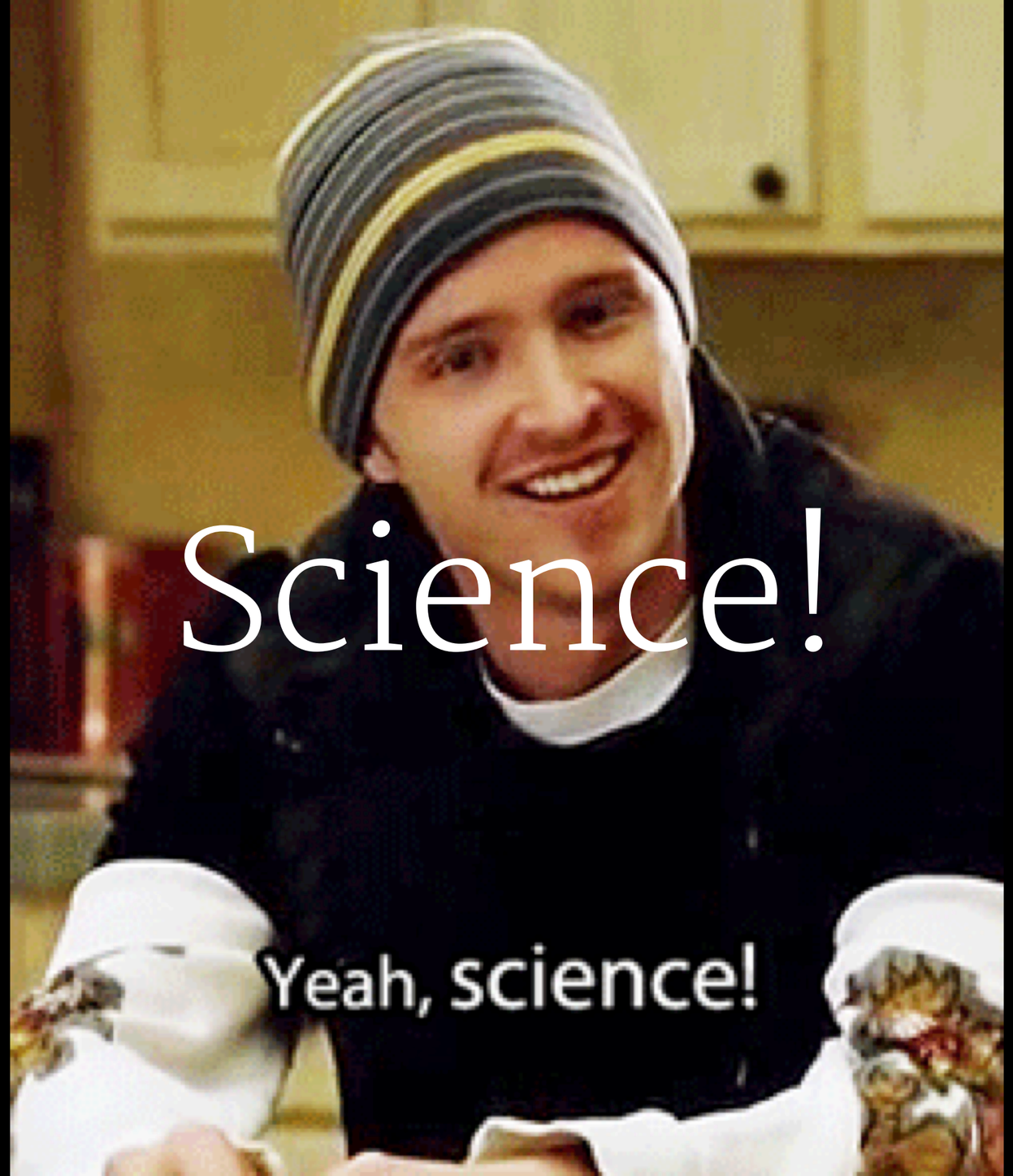
- Launching application
- Building views (Solving Auto Layout constraints)
- Data -> Dictionaries -> Model Objects

CPU

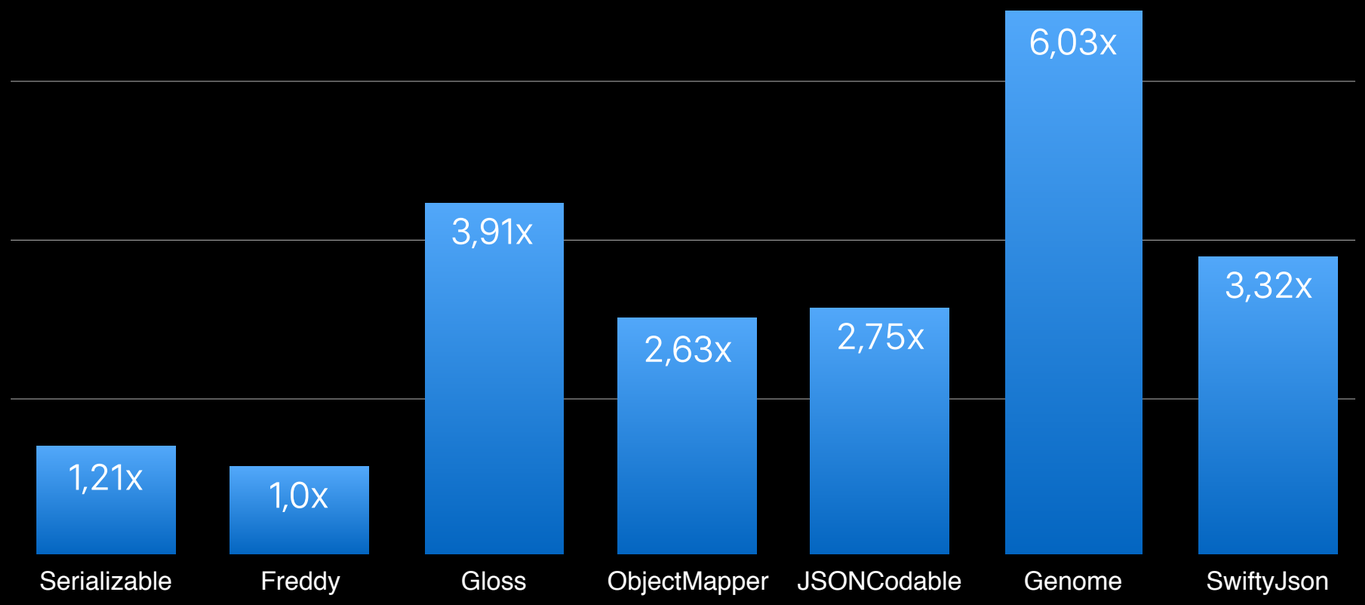
- Launching application 🙈🙉
- Building views (Solving Auto Layout constraints)
- Data -> Dictionaries -> Model Objects

CPU

- Launching application 🙈🙉
- Building views (Solving Auto Layout constraints) 🙈🙉
- Data -> Dictionaries -> Model Objects


Science!
Yeah, science!

CPU – Parsing time



~40.000 lines of JSON

CPU

- Identify CPU heavy activities
- Try to optimize it!

Energy

Utilization

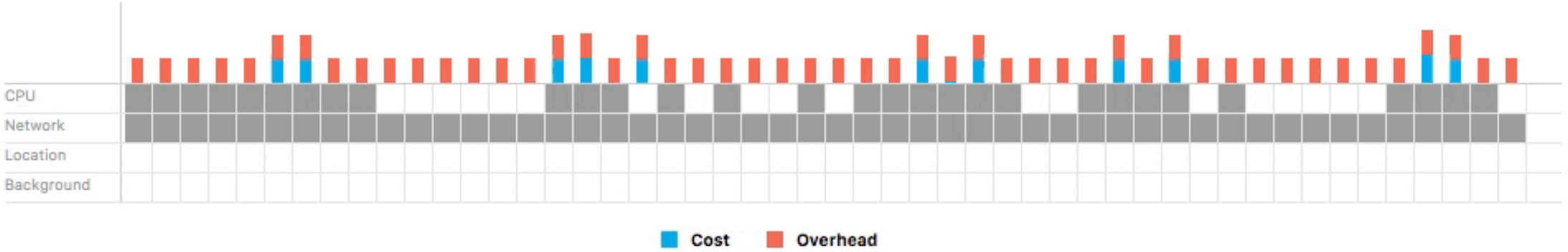
High
Energy Impact

Average

High
Energy Impact

66%
Overhead

Energy Impact



Networking – Apple recommends

- Request servers when needed. Avoid polling!
- Transmit the smallest amount of data needed.
- Transmit data in bursts rather than spreading out transmission packets over time.

Networking – Apple recommends

- Request servers when needed. Avoid polling! 🙈🙉
- Transmit the smallest amount of data needed.
- Transmit data in bursts rather than spreading out transmission packets over time.

Networking – Apple recommends

- Request servers when needed. Avoid polling! 🙈🙉
- Transmit the smallest amount of data needed. 🙈🙉
- Transmit data in bursts rather than spreading out transmission packets over time.

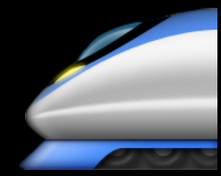
Networking – Apple recommends

- Request servers when needed. Avoid polling! 🙈🙉
- Transmit the smallest amount of data needed. 🙈🙉
- Transmit data in bursts rather than spreading out transmission packets over time.👌

Network polling

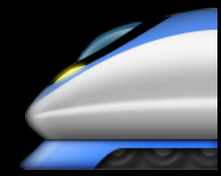
- Data feeds polling independently
- Data feeds polling with different interval

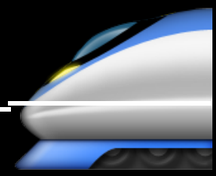
Solution

Trains 

Solution

- Model *work to be done* as a **passenger**
- Schedule passengers into train departures
- Try to satisfy the passengers wishes to departure time while not doing too many departures

Trains 

~~Trains~~ 

UberPOOL 

Algorithm

(no passengers)

Algorithm

🐱 - 30 min (🚗)

Algorithm

🐱 - 30 min

🐶 - 10 min (🚗)

Algorithm

🐱 - 30 min

🐶 - 10 min (🚗)

🐭 - 12 min (🚗)

Algorithm

🐱 - 27 min
🐶 - 7 min (🚗)
🐭 - 7 min (🚗)

Algorithm

🐱 - 27 min
🐶 - 7 min (🚗)
🐭 - 7 min (🚗)
🐼 - 5 min (🚗)

Algorithm

🐱 - 24 min
🐶 - 4 min (🚗)
🐭 - 4 min (🚗)
🐼 - 2 min (🚗)

Algorithm

🚗- -

🐱 - 20 min

🐶 - 0 min (🚗)

🐭 - 0 min (🚗)

🐼 - 0 min (🚗)

Algorithm

🐱 - 20 min

Algorithm

---🚗---

🐱 - 20 min (🚗)

Algorithm

🐱 - 20 min

🐻 - 10 min (🚗)

Algorithm

🐱 - 16 min

🐻 - 6 min (🚗)

Algorithm

🐱 - 13 min

🐻 - 3 min (🚗)

Algorithm

🚗- -

🐱 - 11 min

🐻 - 1 min (🚗)

🦁 - 0 min (🚗)

Algorithm

--------🚗----------

🐱 - 10 min (🚗)

Algorithm

🐱 - 10 min (🚗)
🐷 - 10 min (🚗)

Algorithm

(no passengers)

Energy

Utilization

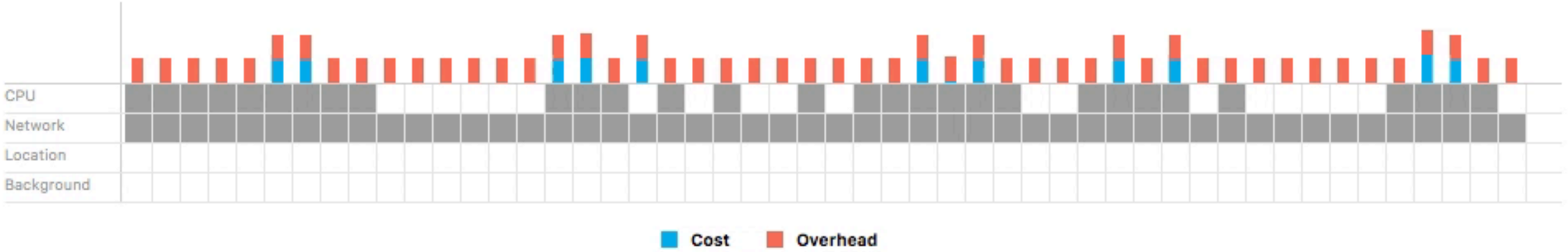
High
Energy Impact

Average

High
Energy Impact

66%
Overhead

Energy Impact



Energy

Utilization

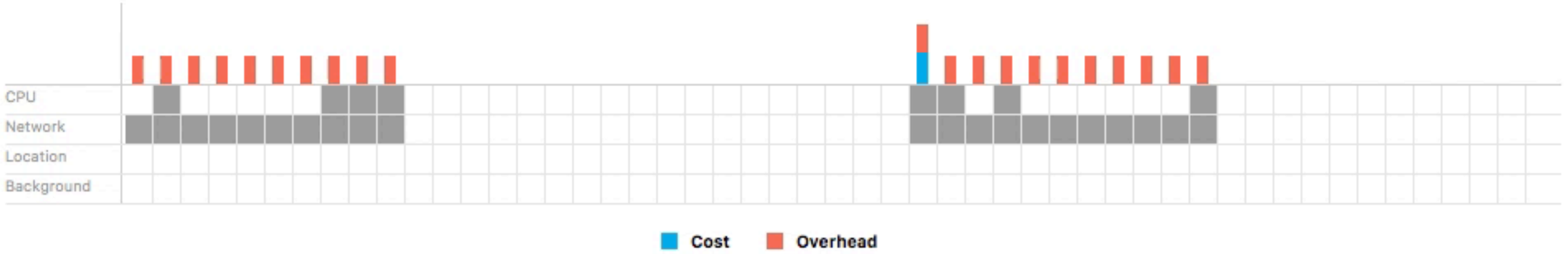
Zero
Energy Impact

Average

Low
Energy Impact

38%
Overhead

Energy Impact



Summary

- Much faster and more efficient parsing of data to model objects
- Batching fetch of data feeds
- Same user experience
- Improved battery efficiency drastically

?