maybe We Should Enable More Uncertain Mobile App Programming

Geoffrey Challen, Jerry Antony Ajay, Nick DiRienzo, Oliver Kennedy,
Anudipa Maiti, Anandatirtha Nandugudi, Sriram Shantharam
Jinghao Shi, Guru Prasad Srinivasa, and Lukasz Ziarek
University at Buffalo

http://blue.cse.buffalo.edu/projects/maybe/

maybe We Should Enable More Uncertain Mobile App Programming

Geoffrey Challen, Jerry Antony Ajay, Nick DiRienzo, Oliver Kennedy,
Anudipa Maiti, Anandatirtha Nandugudi, Sriram Shantharam
Jinghao Shi, Guru Prasad Srinivasa, and Lukasz Ziarek
University at Buffalo

http://blue.cse.buffalo.edu/projects/maybe/

```
String presenter = maybe "Ali Ben Ali", "Jerry Ajay";
```

Adaptation Today

```
if (batteryLevel < threshold) {
  // Sacrifice performance to save energy
} else {
  // Don't sacrifice performance
}
```

Adaptation Today

One threshold?

```
if (batteryLevel < threshold) {
  // Sacrifice performance to save energy
} else {
  // Don't sacrifice performance
}
```

Adaptation Today

One threshold?

What threshold?

```
if (batteryLevel < threshold) {
  // Sacrifice performance to save energy
} else {
  // Don't sacrifice performance
}
```

Adaptation Today

One threshold?

What threshold?

```
if (batteryLevel < threshold) {
  // Sacrifice performance to save energy
} else {
  // Don't sacrifice performance
}
```

Do the branches even work?

Adaptation Today

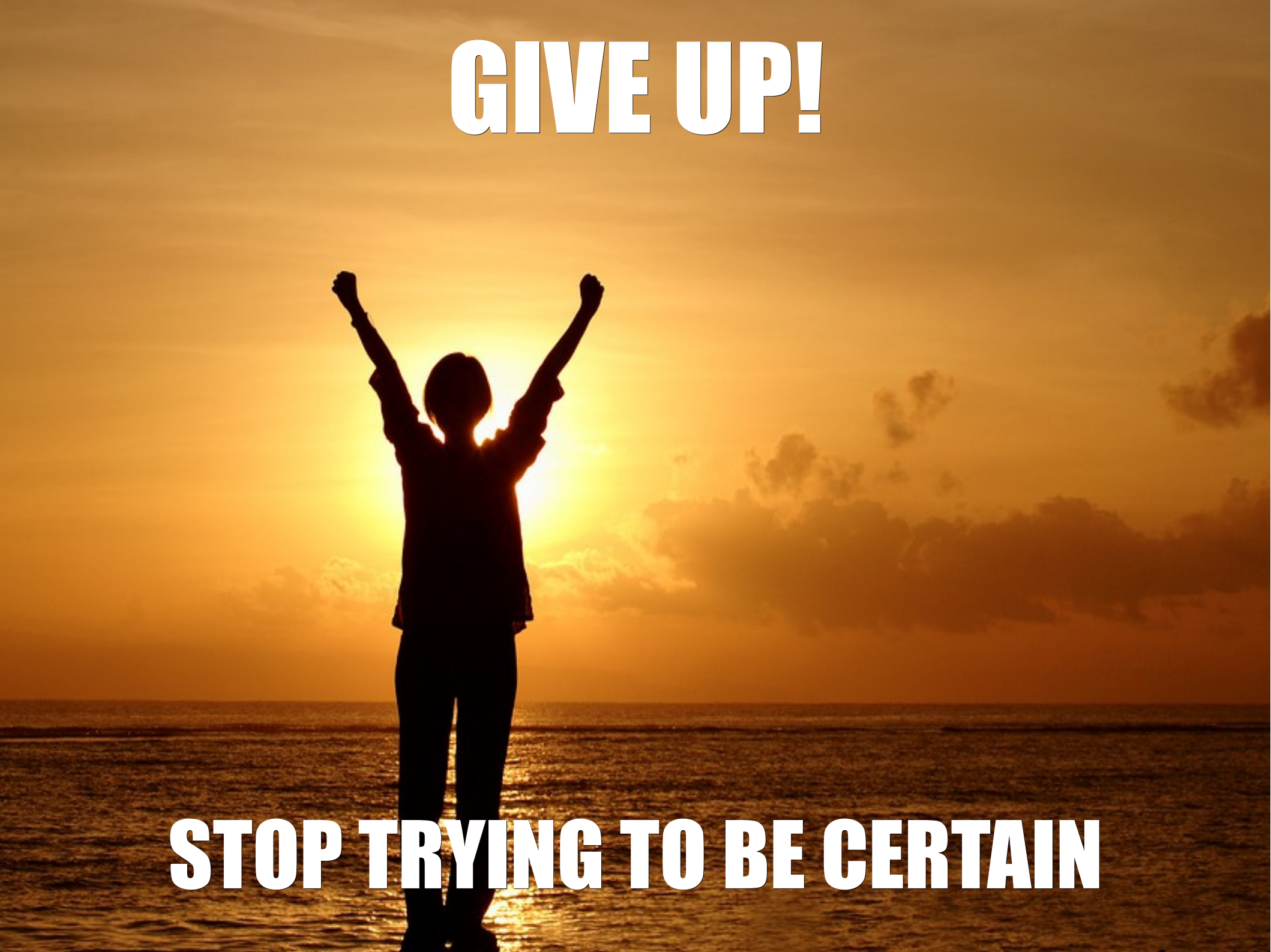
One threshold?

What threshold?

```
if (batteryLevel < threshold) {
  // Sacrifice performance to save energy
} else {
  // Don't sacrifice performance
}
```

And if so, how well?

Do the branches even work?


GIVE UP!
STOP TRYING TO BE CERTAIN

Adaptation Today

```
if (batteryLevel < threshold) {
  // Sacrifice performance to save energy
} else {
  // Don't sacrifice performance
}
```

Adaptation Tomorrow

```
maybe {
   // One way of doing something
} or {
   // Another way
}
```

`maybe` Block Syntax

```
maybe (“label”) {
  // Alternative 1
} or {
  // Alternative 2
}
...
} or {
  // Alternative n
}
```

`maybe` Block Syntax

```
maybe (“label”) {
   // Alternative 1
} or {
   // Alternative 2
}
...
} or {
   // Alternative n
}
```

maybe Block Syntax

Label identifying the statement to the system

```
maybe (“label”) {
   // Alternative 1
} or {
   // Alternative 2
}
...
} or {
   // Alternative n
}
```

Multiple alternatives

`maybe` Assignment Syntax

```
String perf = maybe (“perf”) “low”, ”med”, ”high”;
```

`maybe` Assignment Syntax

Multiple alternatives

```
String perf = maybe (“perf”) “low”, ”med”, ”high”;
```

`maybe` Assignment Syntax

Multiple alternatives

```
String perf = maybe (“perf”) “low”, ”med”, ”high”;

int timer = maybe (“timer”) 1-16;
```

Range shortcut

`maybe` Syntax and System

`maybe` statement: allows programmers to *express uncertainty*

`maybe` Syntax and System

`maybe` statement: allows programmers to *express uncertainty*

`maybe` system: uses testing and learning to *convert uncertainty to runtime certainty*

`maybe` Syntax and System

`maybe` statement: allows programmers to *express uncertainty*

`maybe` system: uses testing and learning to *convert uncertainty to runtime certainty*

Goal: know the best alternative at runtime

maybe Outcomes

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```

`maybe` Outcomes: Static

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```

User driven

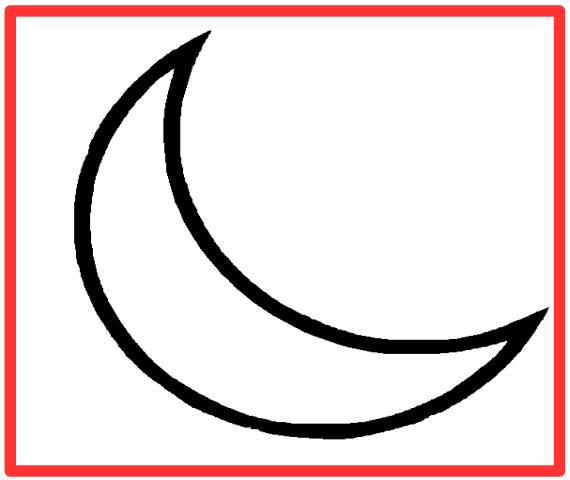
`maybe` Outcomes: Static

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```

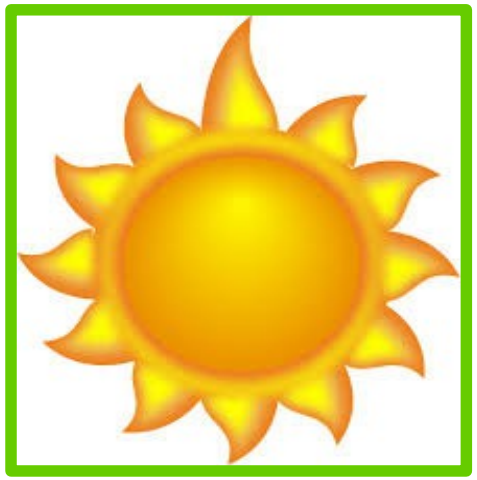
Geographic

`maybe` Outcomes: Dynamic

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```



Temporal



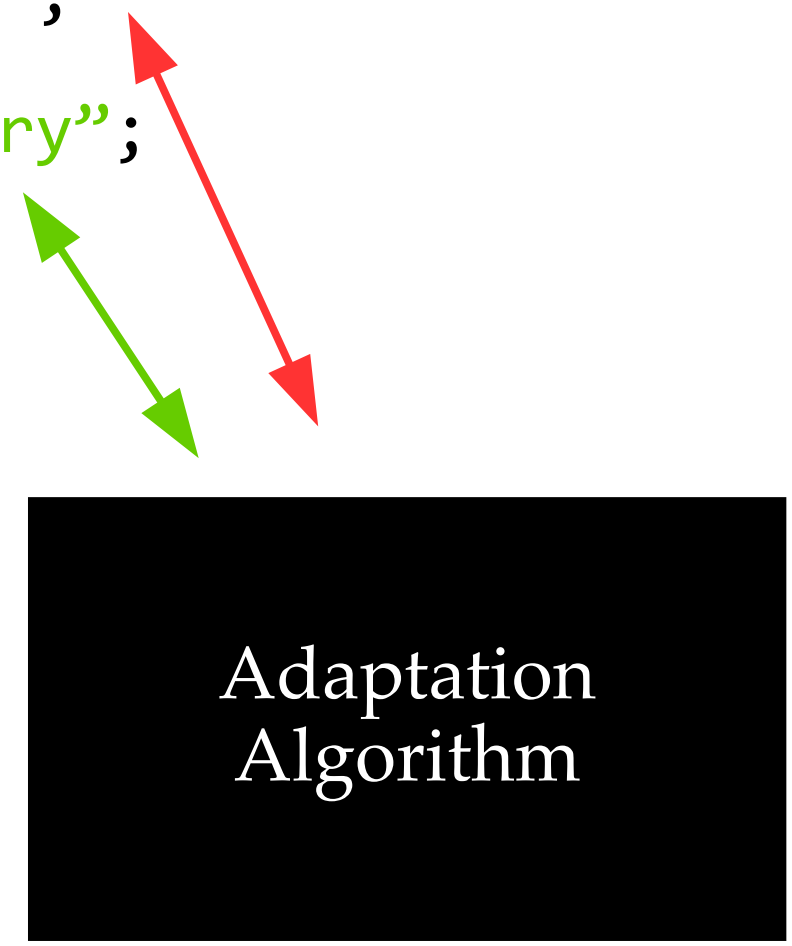
maybe Outcomes: Dynamic

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```

Network based

maybe Outcomes: Dynamic

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```



`maybe` Outcomes: Dynamic

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```

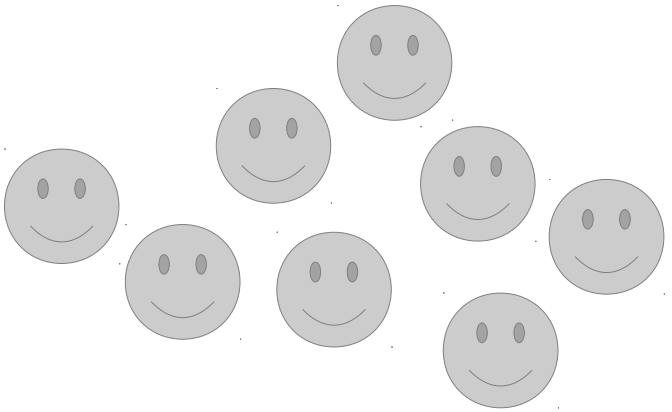
Developer controlled

maybe Testing

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```

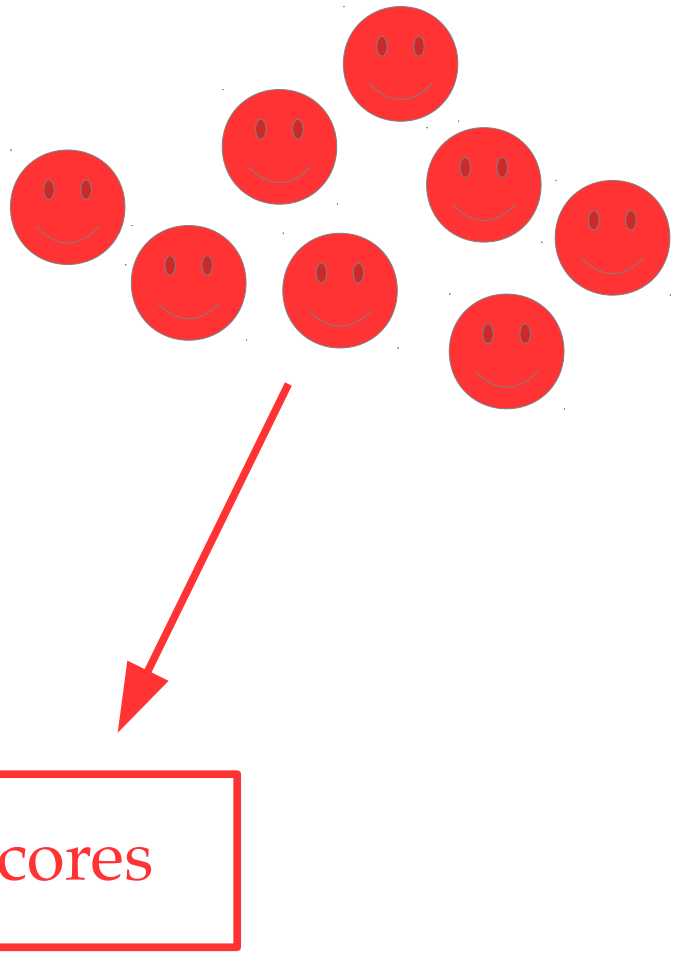
maybe Testing

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```



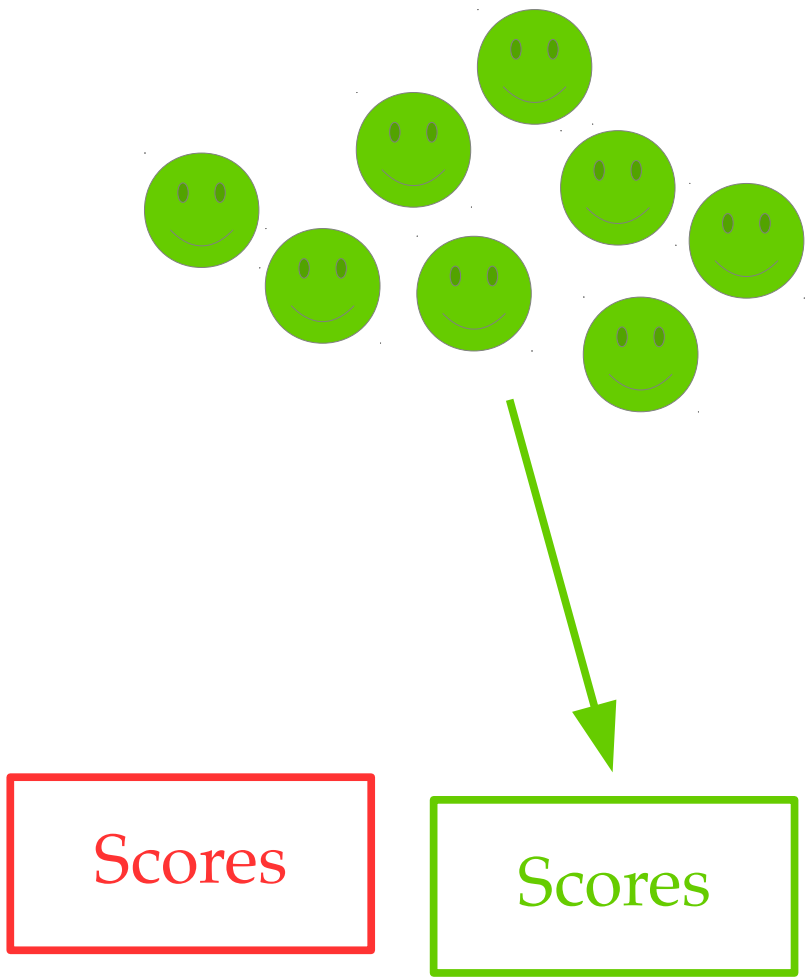
maybe Testing

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```



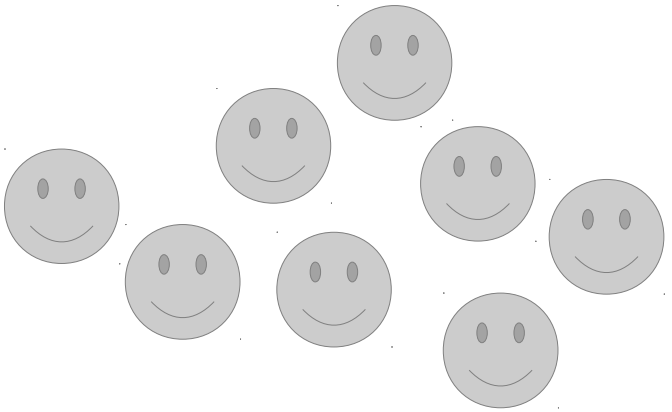
maybe Testing

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```



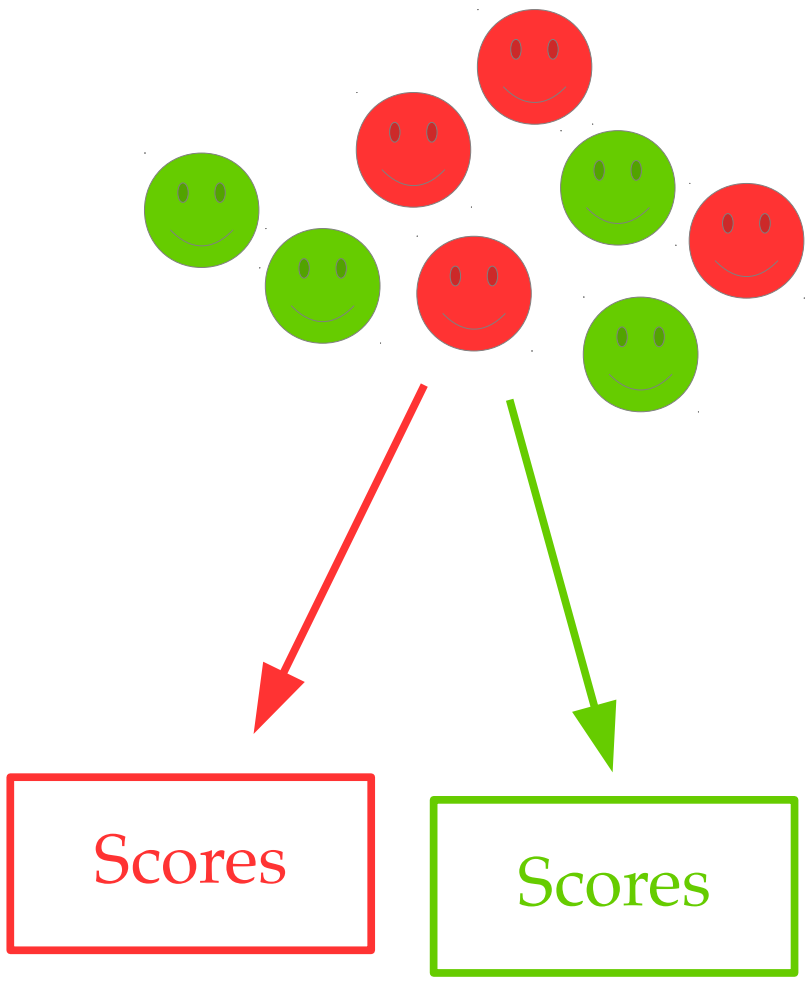
maybe Testing

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```



maybe Testing

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```



maybe Scoring: Standard

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```

`maybe` Scoring: Standard

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```

Performance
Energy consumption

maybe Scoring: Custom

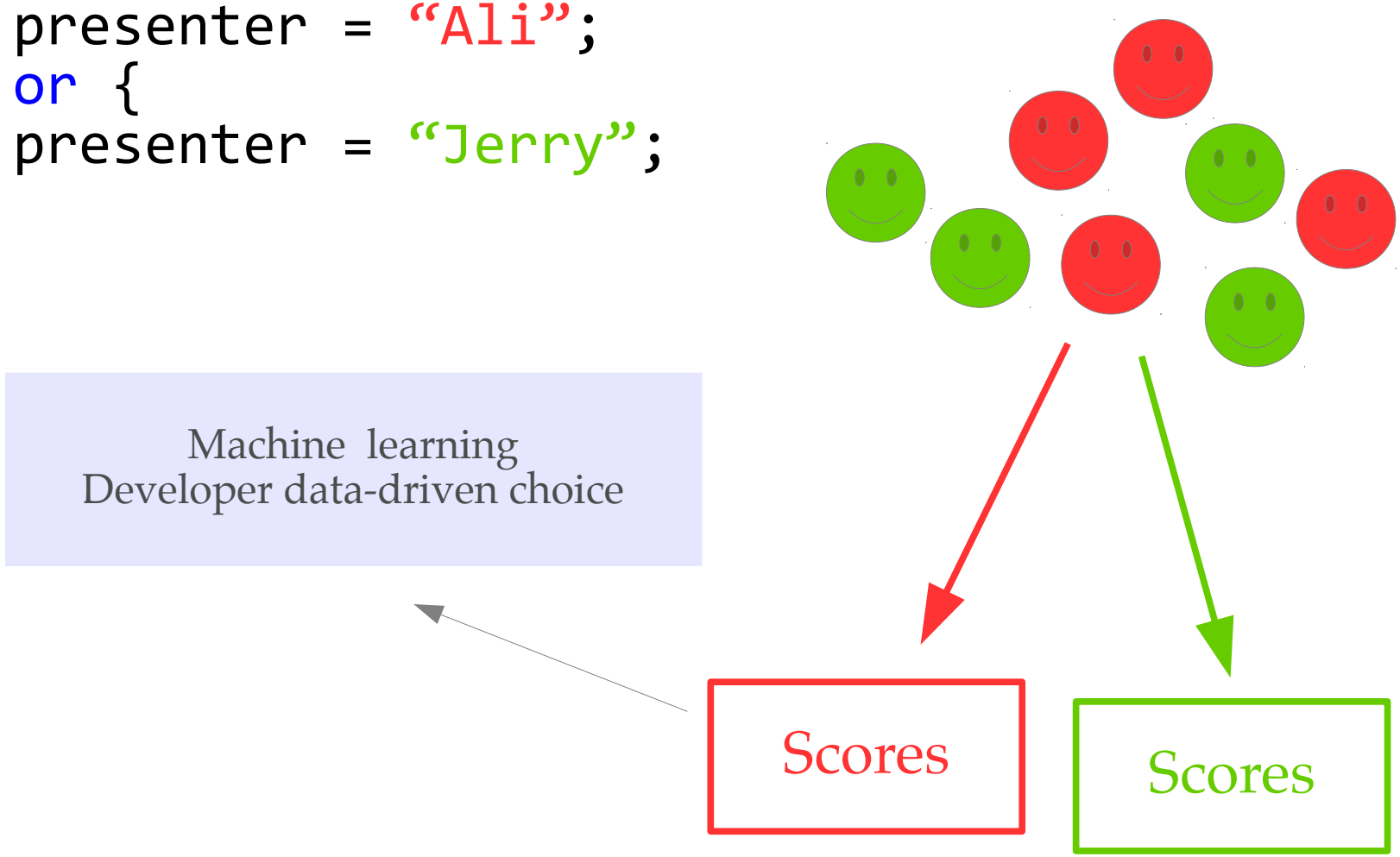
```
// Custom scoring initialization

maybe (“presenter”) {
  presenter = “Ali”;
} or {
  presenter = “Jerry”;
}

maybeService.addScore(“presenter”, JSON);
```

maybe Learning

```
maybe (“presenter”) {
   presenter = “Ali”;
} or {
   presenter = “Jerry”;
}
```



Inspiration: PhoneLab Data Collection

PhoneLab data collection tool

Inspiration: PhoneLab Data Collection

PhoneLab data collection tool

- Multiple sources of uncertainty:
  - When and how often to upload data?
  - How often to rotate logs?
  - How much data to cache on the smartphone?

Inspiration: PhoneLab Data Collection

PhoneLab data collection tool

- Multiple sources of uncertainty:
  - When and how often to upload data?
  - How often to rotate logs?
  - How much data to cache on the smartphone?
- End-to-end performance metrics

Inspiration: PhoneLab Data Collection

PhoneLab data collection tool

- Multiple sources of uncertainty:
  - When and how often to upload data?
  - How often to rotate logs?
  - How much data to cache on the smartphone?
- End-to-end performance metrics
- Wrote a large deal of boring boilerplate code to implement app-specific adaptation

Inspiration: PhoneLab Data Collection

PhoneLab data collection tool

- Multiple sources of uncertainty:
  - When and how often to upload data?
  - How often to rotate logs?
  - How much data to cache on the smartphone?
- End-to-end performance metrics
- Wrote a large deal of boring boilerplate code to implement app-specific adaptation

```
int uploadMin = maybe (“upload”) 30, 60, 120;
```

Inspiration: Parking Lot Monitoring

PocketParker parking lot monitoring app

Inspiration: Parking Lot Monitoring

PocketParker parking lot monitoring app

- Relied on detecting transitions between walking and driving

Inspiration: Parking Lot Monitoring

PocketParker parking lot monitoring app

- Relied on detecting transitions between walking and driving
- Originally wrote our own activity-recognition code

Inspiration: Parking Lot Monitoring

PocketParker parking lot monitoring app

- Relied on detecting transitions between walking and driving
- Originally wrote our own activity-recognition code
- ... then Google released their own library.

Inspiration: Parking Lot Monitoring

PocketParker parking lot monitoring app

- Relied on detecting transitions between walking and driving
- Originally wrote our own activity-recognition code
- ... then Google released their own library.
- Which is better?

Inspiration: Parking Lot Monitoring

PocketParker parking lot monitoring app

- Relied on detecting transitions between walking and driving
- Originally wrote our own activity-recognition code
- ... then Google released their own library.
- Which is better?

```
maybe (“algorithm”) {
  initializeCustomAlgorithm();
} or {
  initializeGoogleAlgorithm();
}
```

Mining for Uncertainty

Currently we are forced to use `maybe` to explore "hidden uncertainty" in code forced to be certain

Mining for Uncertainty

Currently we are forced to use `maybe` to explore "hidden uncertainty" in code forced to be certain.

Uncertainty hints:

- Timer rates
- Timeouts
- Cache sizes
- Performance-quality tradeoffs
- Attempts at battery level adaptation
- Any suggestions?

Current `maybe` Prototype

Rewrite-based implementation for Android Java

Current `maybe` Prototype

Rewrite-based implementation for Android Java

- Preprocessor to rewrite `maybe` to existing conditional statements

Current `maybe` Prototype

Rewrite-based implementation for Android Java

- Preprocessor to rewrite `maybe` to existing conditional statements
- Android service to cache values and implement adaptation algorithms

Current `maybe` Prototype

Rewrite-based implementation for Android Java

- Preprocessor to rewrite `maybe` to existing conditional statements
- Android service to cache values and implement adaptation algorithms
- Backend server to drive testing, collect data and perform data analysis and learning

Current `maybe` Prototype

Rewrite-based implementation for Android Java

- Preprocessor to rewrite `maybe` to existing conditional statements
- Android service to cache values and implement adaptation algorithms
- Backend server to drive testing, collect data and perform data analysis and learning
- Web interface to allow programmers to control the process

Summary: Embrace Uncertainty

- Forced certainty is a problem—particularly for mobile systems
- The `maybe` statement and system allow programmers to express and embrace uncertainty and activate testing and machine learning techniques

```
maybe (“findOutMore”) {
  http://blue.cse.buffalo.edu/projects/maybe
} or {
  http://www.stupidest_idea_ever.com
}
```