
Google™
11
I/O

Beyond JavaScript: Programming the Web with Native Client

David Springer

Agenda

Agenda

- Technical overview
  - Merge two worlds: web and client applications
  - Bring native code to the web

Agenda

- Technical overview
  - Merge two worlds: web and client applications
  - Bring native code to the web
- Use cases
  - 90/10 and 10/90 applications

Agenda

- Technical overview
  - Merge two worlds: web and client applications
  - Bring native code to the web
- Use cases
  - 90/10 and 10/90 applications
- Life!
  - Build an app

Agenda

- Technical overview
  - Merge two worlds: web and client applications
  - Bring native code to the web
- Use cases
  - 90/10 and 10/90 applications
- Life!
  - Build an app
- Q&A

What Is NaCl?

What Is NaCl?

- “Native Client” <=> “NaCl”

What Is NaCl?

- “Native Client” <=> “NaCl”
- Sandboxing technology for safe* execution of native code
  - Native machine code run on the client CPU
  - Not an interpreter
  - Platform independent: POSIX-like API, Pepper

What Is NaCl?

- “Native Client” <=> “NaCl”
- Sandboxing technology for safe* execution of native code
  - Native machine code run on the client CPU
  - Not an interpreter
  - Platform independent: POSIX-like API, Pepper
- Merge web and client application benefits
  - “Write once run everywhere” like a web app
  - Run native code like a client app

What Is NaCl?

- “Native Client” <=> “NaCl”
- Sandboxing technology for safe* execution of native code
  - Native machine code run on the client CPU
  - Not an interpreter
  - Platform independent: POSIX-like API, Pepper
- Merge web and client application benefits
  - “Write once run everywhere” like a web app
  - Run native code like a client app
- Restrictions based on security

How Does NaCl Work?

(Un)Trust and Verify

How Does NaCl Work?

(Un)Trust and Verify

- Compile code using specialized toolchain
  - Builds for many CPU architectures

How Does NaCl Work?

(Un)Trust and Verify

- Compile code using specialized toolchain
  - Builds for many CPU architectures
- NaCl runtime verifies untrusted code
  - Static analysis

How Does NaCl Work?

(Un)Trust and Verify

- Compile code using specialized toolchain
  - Builds for many CPU architectures
- NaCl runtime verifies untrusted code
  - Static analysis
- Execute verified code
  - NaCl is not an interpreter

How Does NaCl Work?

(Un)Trust and Verify

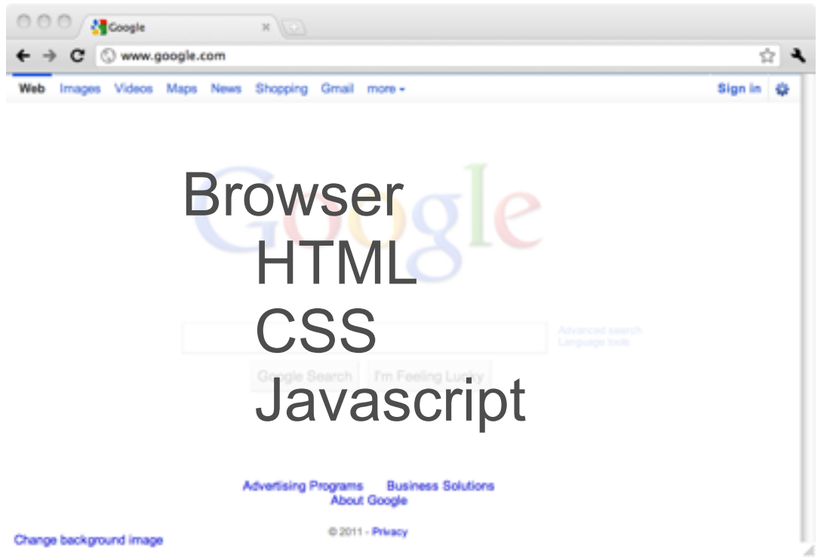
- Compile code using specialized toolchain
  - Builds for many CPU architectures
- NaCl runtime verifies untrusted code
  - Static analysis
- Execute verified code
  - NaCl is not an interpreter
- Halt code if it tries naughty stuff
  - Can't touch the file system: `open()`, `getcwd()`, `stat()`, etc.
  - No process control: `fork()`, `exec()`, `kill()`, etc.
  - For a complete list: `nacl-nm nacl/lib/libnosys.a`

How Does NaCl Work?

Play in the Sandbox

How Does NaCl Work?

Play in the Sandbox

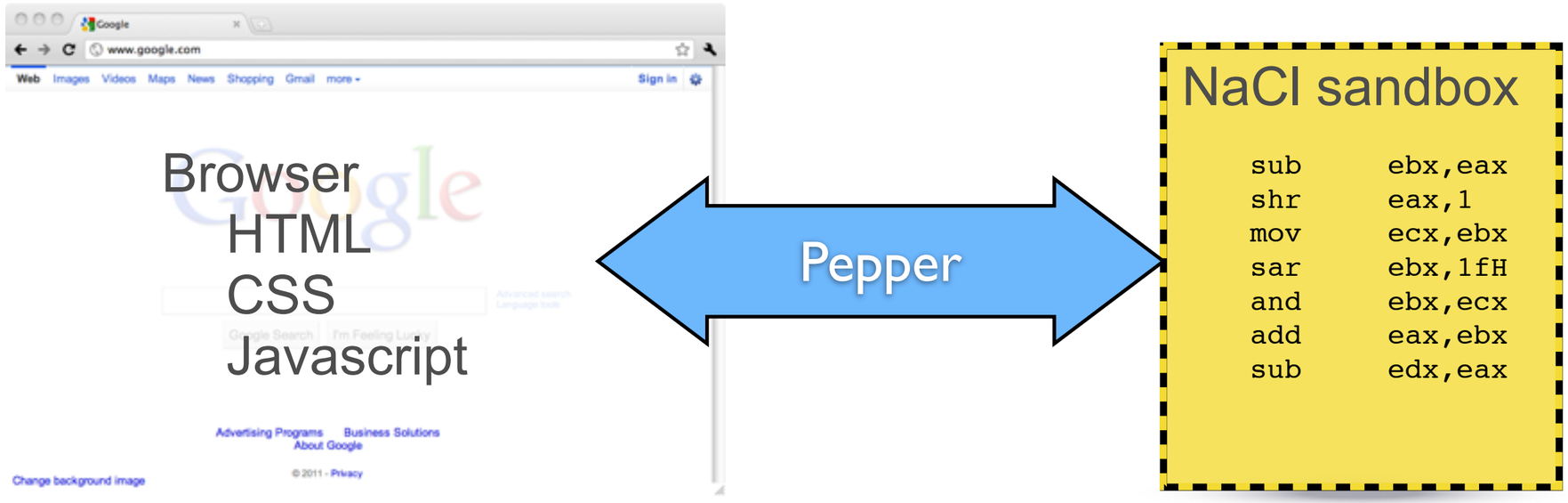


How Does NaCl Work?

Play in the Sandbox

How Does NaCl Work?

Play in the Sandbox



How Does NaCl Work?

Spot the Evil Code

```
mov     eax, [esi + 0x114]
cmp     eax, edi
jz      $label
mov     ecx, [eax]
push    eax
call    [ecx]
```

How Does NaCl Work?

Spot the Evil Code

```
mov     eax, [esi + 0x114]
cmp     eax, edi
jz      $label
mov     ecx, [eax]
push    eax
call    [ecx]
```

How Does NaCl Work?

Spot the Evil Code

```
mov     eax, [esi + 0x114]
cmp     eax, edi
jz      $label
mov     ecx, [eax]
push    eax
call    [ecx]
```

How Does NaCl Work?

Spot the Evil Code

```
mov     eax, [esi + 0x114]
cmp     eax, edi
jz      $label
mov     ecx, [eax]
push    eax
call    [ecx]
```

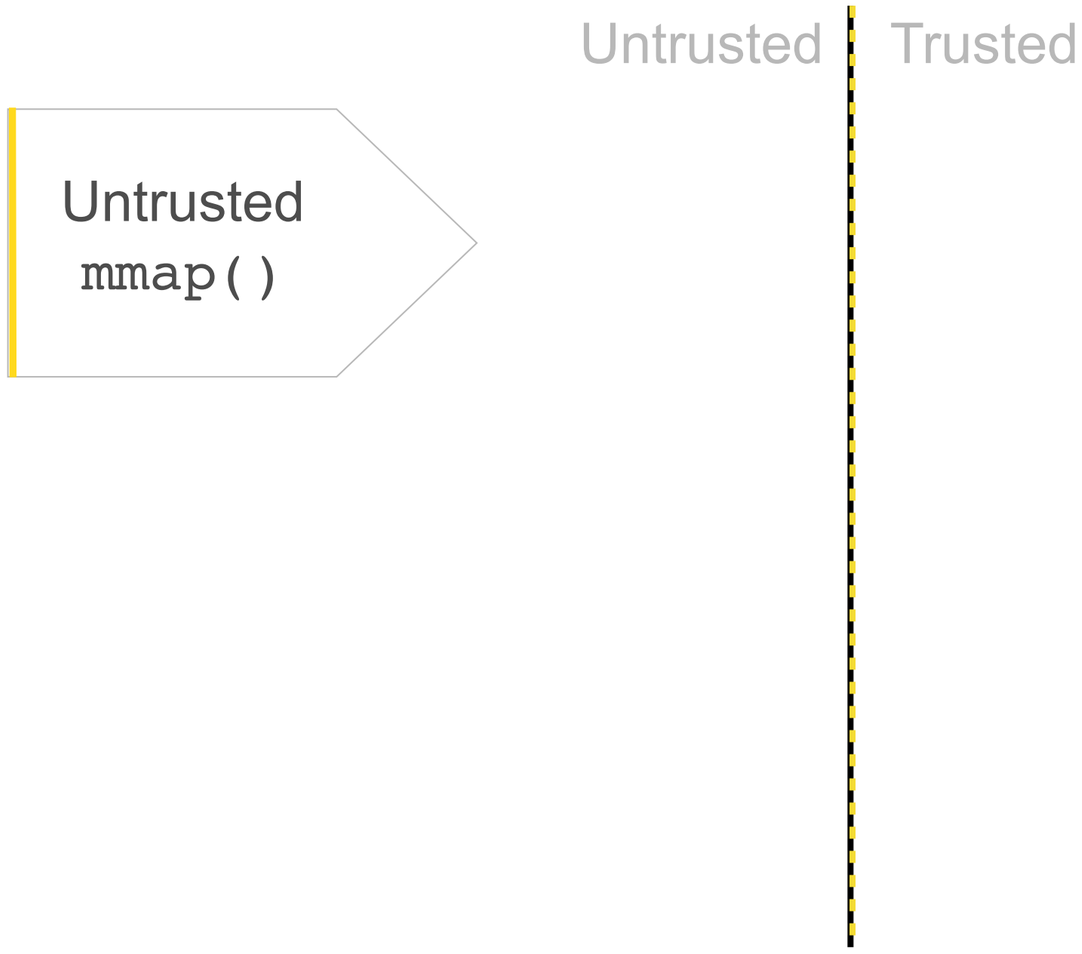
How Does NaCl Work?

Spot the Evil Code

```
mov     ecx, [eax]
push    eax
nop
nop
nop
and     ecx, -32
call    [ecx]
```

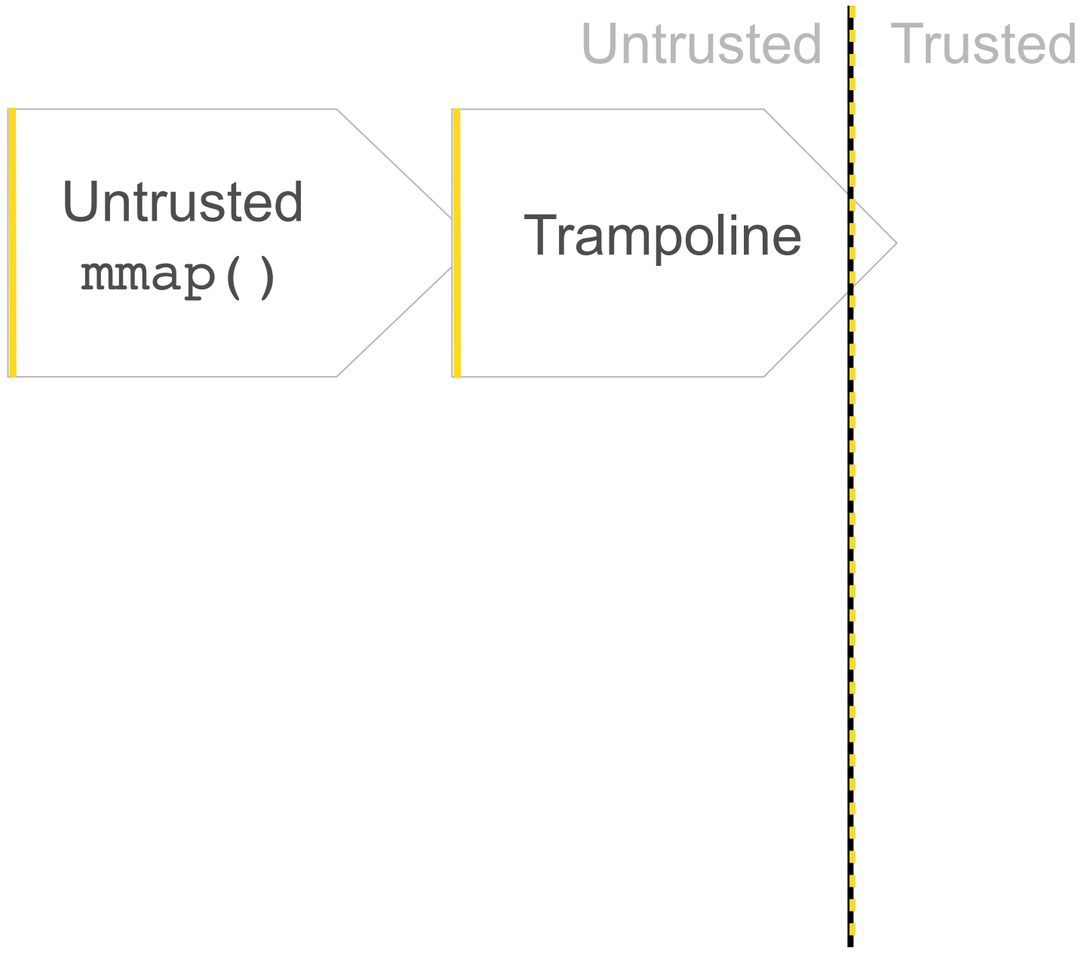
How Does NaCl Work?

The Trampoline



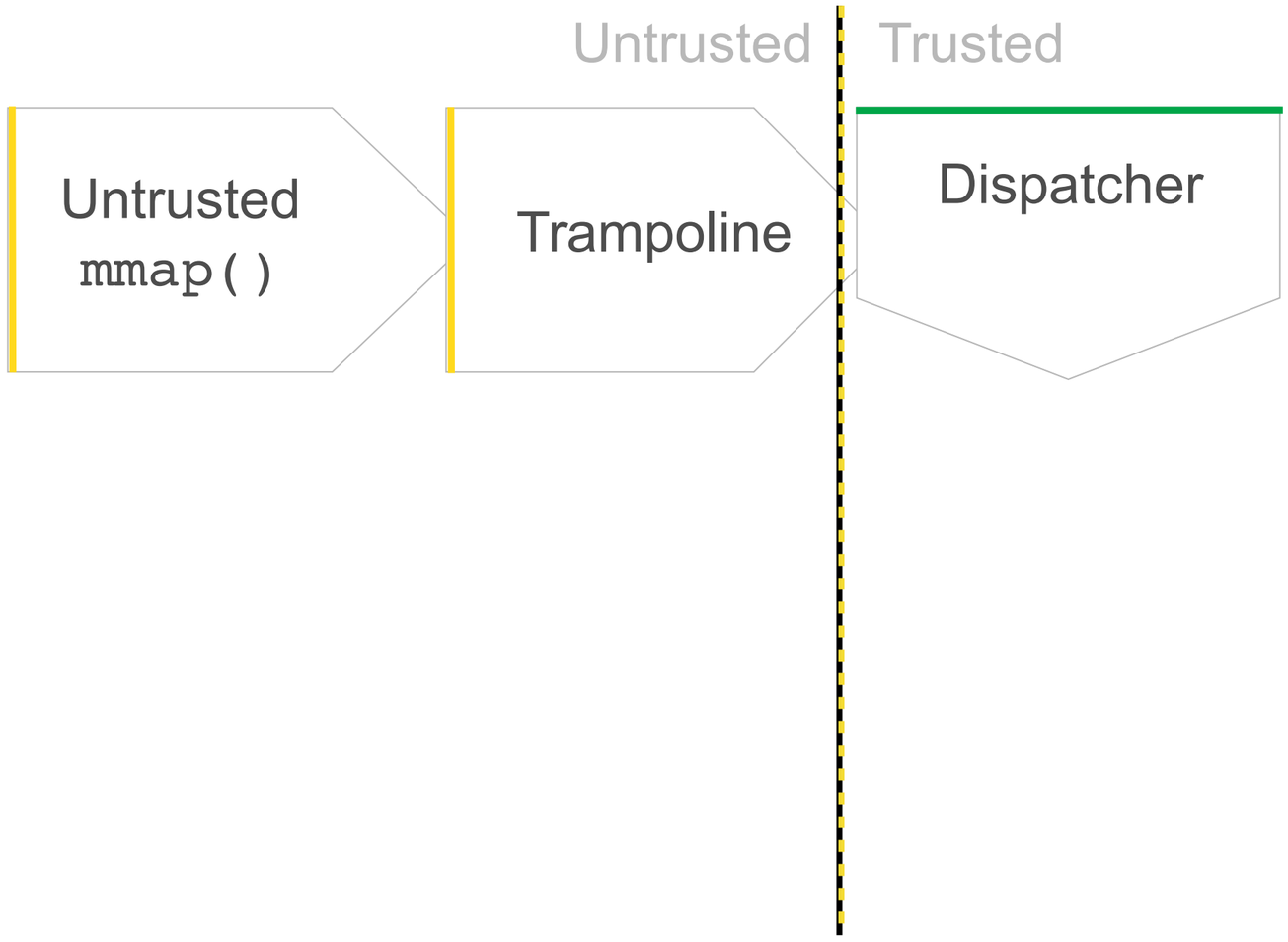
How Does NaCl Work?

The Trampoline



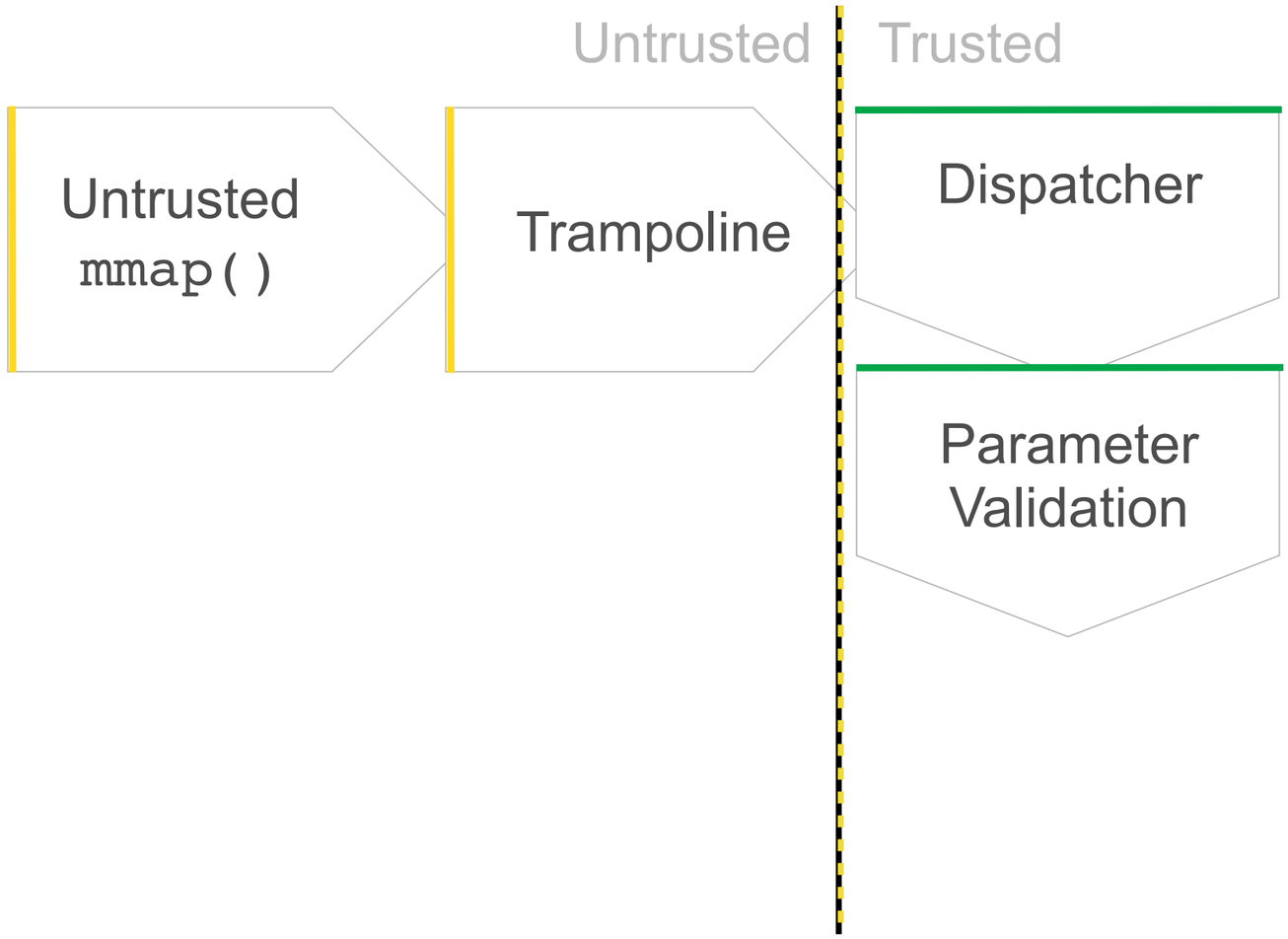
How Does NaCl Work?

The Trampoline



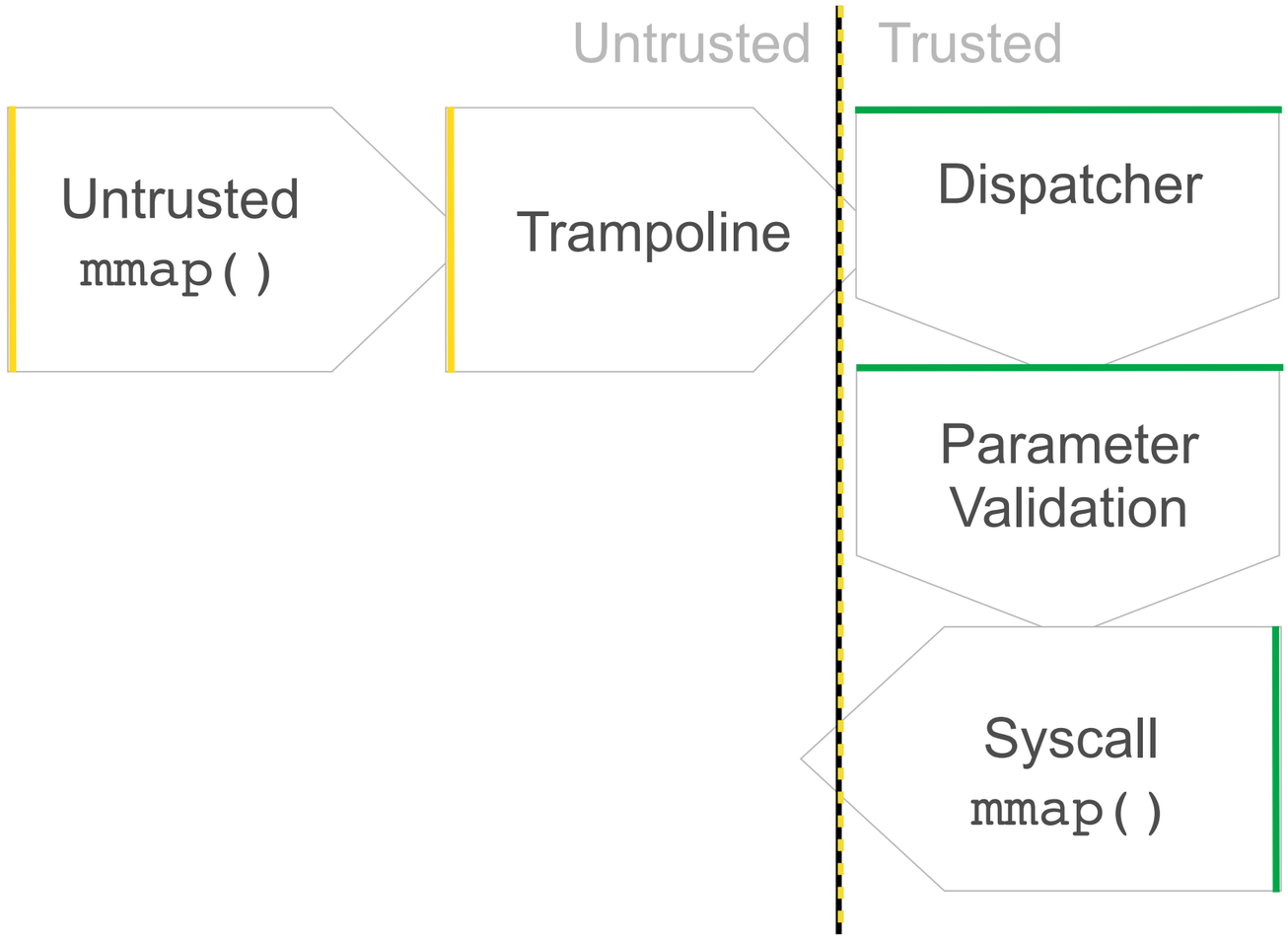
How Does NaCl Work?

The Trampoline



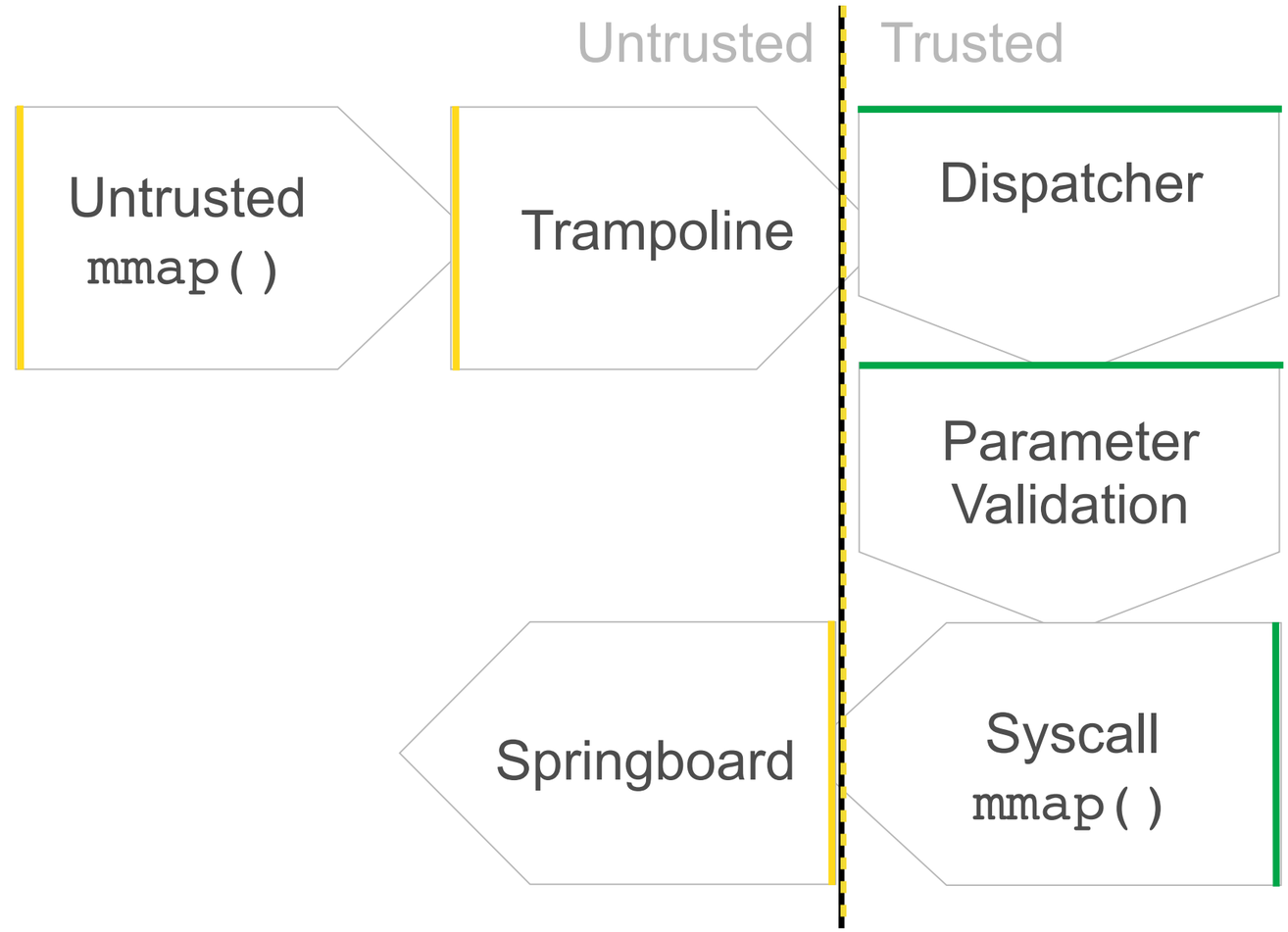
How Does NaCl Work?

The Trampoline



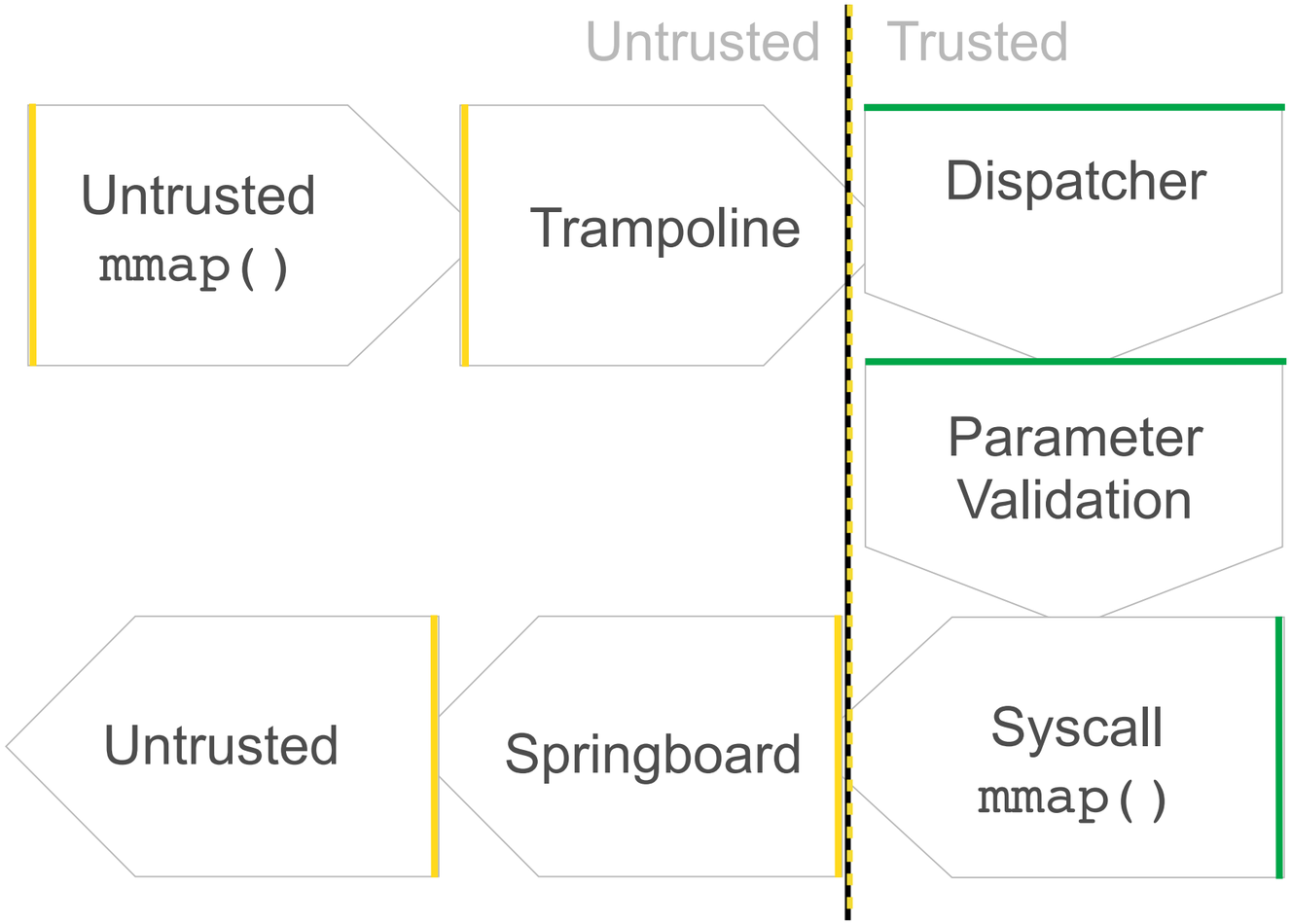
How Does NaCl Work?

The Trampoline



How Does NaCl Work?

The Trampoline



How Does NaCl Work?

Security Model: Suspenders *and* a Belt

- Defense in depth
  - NaCl sandbox and Chrome sandbox
  - CORS
  - CPU blacklisting
  - NaCl module blacklisting
  - Security audits and contests
  - Public review

Why Use NaCl?

Why Use NaCl?

- Language choice
  - C++ -> Javascript? … maybe not …

Why Use NaCl?

- Language choice
  - C++ -> Javascript? … maybe not …
- Performance
  - Direct (sandboxed) CPU and GPU access
  - SIMD instructions and multicore CPUs
  - Full control over layout of code and data

Why Use NaCl?

- Language choice
  - C++ -> Javascript? … maybe not …
- Performance
  - Direct (sandboxed) CPU and GPU access
  - SIMD instructions and multicore CPUs
  - Full control over layout of code and data
- Browser as an application platform
  - Ease of application delivery
  - No user install
  - Chrome Web Store

Why Use Na

- Language ch
  - C++ -> Javas
- Performance
  - Direct (sandb
  - SIMD instruct
  - Full control ov
- Browser as a
  - Ease of applic
  - No user install
  - Chrome Web Store

Confirm Installation

This webpage wants to install software from MightNotBeEvil, Inc.

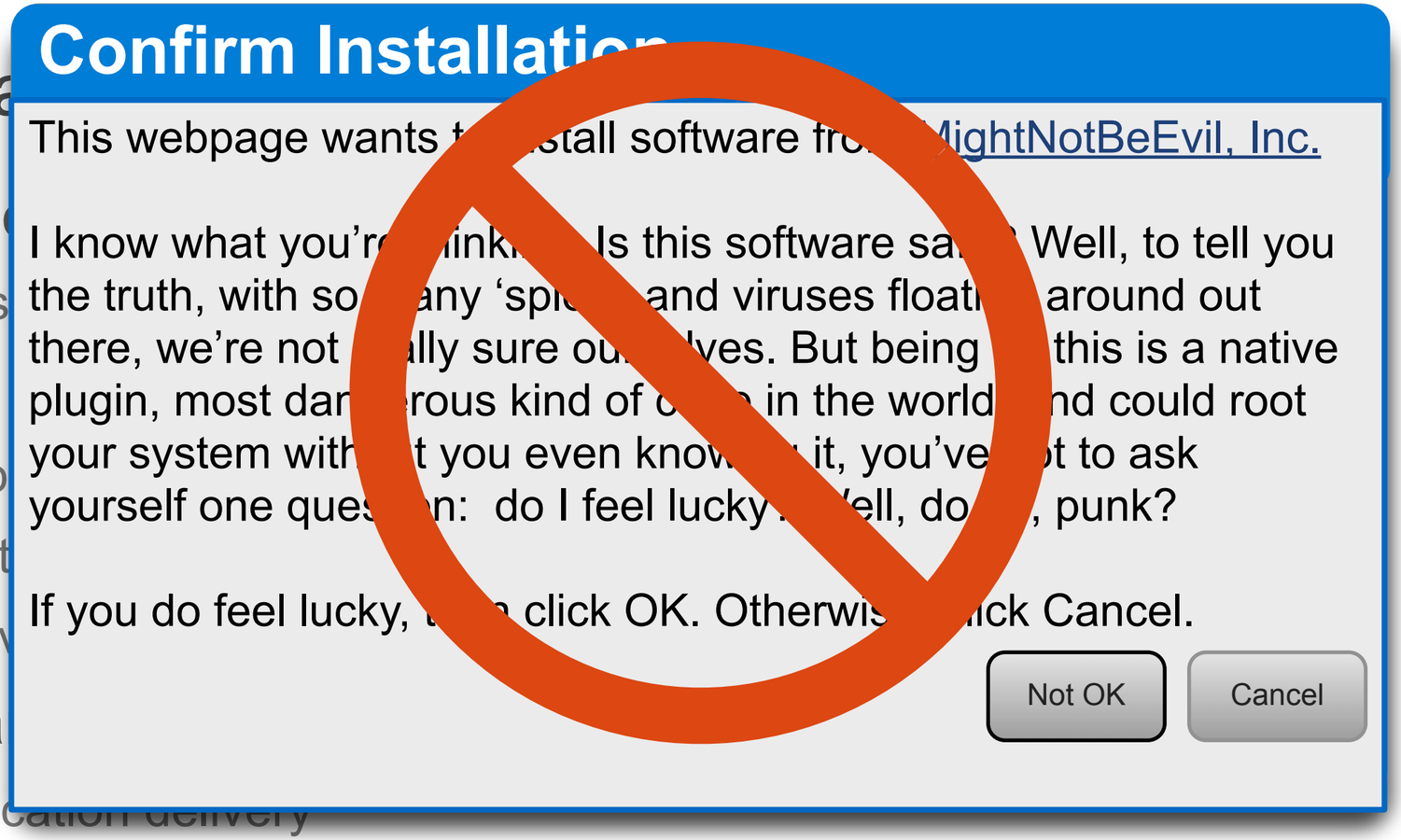
I know what you're thinking. Is this software safe? Well, to tell you the truth, with so many 'sploits and viruses floating around out there, we're not really sure ourselves. But being as this is a native plugin, most dangerous kind of code in the world, and could root your system without you even knowing it, you've got to ask yourself one question: do I feel lucky? Well, do ya, punk?

If you do feel lucky, then click OK. Otherwise, click Cancel.

Not OK | Cancel

Why Use Na

- Language ch
  - C++ -> Javas
- Performance
  - Direct (sandb
  - SIMD instruct
  - Full control ov
- Browser as a
  - Ease of applic
  - No user install
  - Chrome Web Store



What is NaCl Good For?

90/10 or 10/90 Applications

What is NaCl Good For?

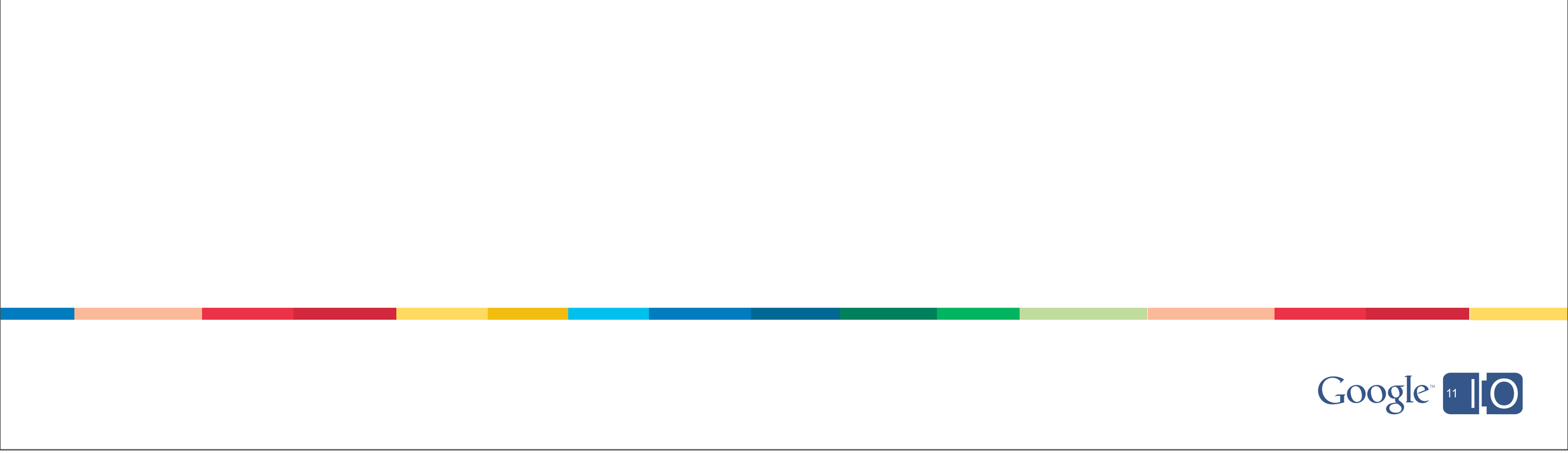
90/10 or 10/90 Applications

- 90/10 applications
  - Web-centric apps that need an engine
  - UI, controller logic in the browser
  - Like having a very low-latency local server
  - Example: local crypto (DRM)
  - Example: physics engine

What is NaCl Good For?

90/10 or 10/90 Applications

- 90/10 applications
  - Web-centric apps that need an engine
  - UI, controller logic in the browser
  - Like having a very low-latency local server
  - Example: local crypto (DRM)
  - Example: physics engine
- 10/90 applications
  - Entire application in native code
  - Example: immersive client games


Google
11

4 Stages of Life

4 Stages of Life

4 Stages of Life

- Get started
  - Build a web site

4 Stages of Life

- Get started
  - Build a web site
- Integrate with the browser
  - Connect Javascript to C++

4 Stages of Life

- Get started
  - Build a web site
- Integrate with the browser
  - Connect Javascript to C++
- Optimize
  - Multi-threading

4 Stages of Life

- Get started
  - Build a web site
- Integrate with the browser
  - Connect Javascript to C++
- Optimize
  - Multi-threading
- Multimedia and web assets

Life Stage 1: Hello, World!

Birth

- `init_project.py`
  - HTML template
  - `.nmf` file
- Build
  - Runs the toolchain
  - Builds for all supported CPU architectures
- `httpd.py`
  - `file:` URLs are not allowed
- Launch Chrome
- `localhost:5103/life2011/life_stage_1/life.html`

Life Stage 1: Hello, World!

Pull Yourself Together

Life Stage 1: Hello, World!

Pull Yourself Together

```
ls ~/Sites/life2011/life_stage_1/
```

Life Stage 1: Hello, World!

Pull Yourself Together

```
ls ~/Sites/life2011/life_stage_1/
   life.html
```

Life Stage 1: Hello, World!

Pull Yourself Together

```
ls ~/Sites/life2011/life_stage_1/
   life.html
   life.nmf
```

Life Stage 1: Hello, World!

Pull Yourself Together

```
ls ~/Sites/life2011/life_stage_1/
   life.html
   life.nmf
   life_x86_32.nexe
   life_x86_64.nexe
   life_arm.nexe
```

Life Stage 1: Hello, World!

Pull Yourself Together

```
ls ~/Sites/life2011/life_stage_1/
   life.html
   life.nmf
   life_x86_32.nexe
   life_x86_64.nexe
   life_arm.nexe
   life.css
   toolbar.css
   toolbar-bg.png
```

Life Stage 1: Hello, World!

The Ties that Bind


```
<div id="life_game_view" class="life-view"></div>
<script type="text/javascript">
  var element = document.getElementById('life_game_view')
  var viewSize = {width: element.offsetWidth || 400,
                  height: element.offsetHeight || 400};
  element.innerHTML = '<embed name="nacl_module"' +
                      ' id="life_view"' +
                      ' src="life.nmf"' +
                      ' type="application/x-nacl"' +
                      ' class="autosize"' +
                      ' width=' + viewSize.width +
                      ' height=' + viewSize.height + ' />'
</script>
```

Life Stage 1: Hello, World!

The Ties that Bind


```
<div id="life_game_view" class="life-view"></div>
<script type="text/javascript">
  var element = document.getElementById('life_game_view')
  var viewSize = {width: element.offsetWidth || 400,
                  height: element.offsetHeight || 400};
  element.innerHTML = '<embed name="nacl_module"' +
                      ' id="life_view"' +
                      ' src="life.nmf"' +
                      ' type="application/x-nacl"' +
                      ' class="autosize"' +
                      ' width=' + viewSize.width +
                      ' height=' + viewSize.height + ' />'
</script>
```

Life Stage 1: Hello, World!

The Ties that Bind


```
<div id="life_game_view" class="life-view"></div>
<script type="text/javascript">
  var element = document.getElementById('life_game_view')
  var viewSize = {width: element.offsetWidth || 400,
                  height: element.offsetHeight || 400};
  element.innerHTML = '<embed name="nacl_module"' +
                      ' id="life_view"' +
                      ' src="life.nmf"' +
                      ' type="application/x-nacl"' +
                      ' class="autosize"' +
                      ' width=' + viewSize.width +
                      ' height=' + viewSize.height + ' />'
</script>
```

Life Stage 1: Hello, World!

The Ties that Bind


```
<div id="life_game_view" class="life-view"></div>
<script type="text/javascript">
  var element = document.getElementById('life_game_view')
  var viewSize = {width: element.offsetWidth || 400,
                  height: element.offsetHeight || 400};
  element.innerHTML = '<embed name="nacl_module"' +
                      ' id="life_view"' +
                      ' src="life.nmf"' +
                      ' type="application/x-nacl"' +
                      ' class="autosize"' +
                      ' width=' + viewSize.width +
                      ' height=' + viewSize.height + ' />'
</script>
```

Life Stage 1: Hello, World!

The Ties that Bind


```
<div id="life_game_view" class="life-view"></div>
<script type="text/javascript">
  var element = document.getElementById('life_game_view')
  var viewSize = {width: element.offsetWidth || 400,
                  height: element.offsetHeight || 400};
  element.innerHTML = '<embed name="nacl_module"' +
                      ' id="life_view"' +
                      ' src="life.nmf"' +
                      ' type="application/x-nacl"' +
                      ' class="autosize"' +
                      ' width=' + viewSize.width +
                      ' height=' + viewSize.height + ' />'
</script>
```

Life Stage 1: Hello, World!

The Ties that Bind


```
<div id="life_game_view" class="life-view"></div>
<script type="text/javascript">
  var element = document.getElementById('life_game_view')
  var viewSize = {width: element.offsetWidth || 400,
                  height: element.offsetHeight || 400};
  element.innerHTML = '<embed name="nacl_module"' +
                      ' id="life_view"' +
                      ' src="life.nmf"' +
                      ' type="application/x-nacl"' +
                      ' class="autosize"' +
                      ' width=' + viewSize.width +
                      ' height=' + viewSize.height + ' />'
</script>
```

Life Stage 1: Hello, World!

The Ties that Bind

life.nmf:

```
{
  "nexes": {
    "x86-64": "life_x86_64.nexe",
    "x86-32": "life_x86_32.nexe",
    "ARM": "life_arm.nexe"
  }
}
```

Life Stage 1: Hello, World!

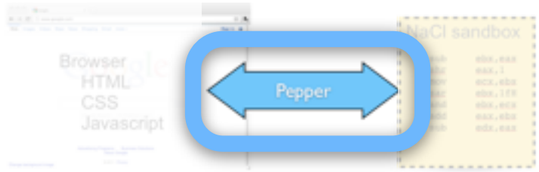
Pepper: The Spice of Life


```
void Life::CreateContext(const pp::Size& size) {
  if (IsContextValid())
    return;
  graphics_2d_context_ = new pp::Graphics2D(this, size, false);
  if (!BindGraphics(*graphics_2d_context_)) {
    printf("Couldn't bind the device context\n");
  }
}

void Life::FlushPixelBuffer() {
  if (!IsContextValid())
    return;
  graphics_2d_context_->PaintImageData(*pixel_buffer_, pp::Point());
  if (flush_pending())
    return;
  set_flush_pending(true);
  graphics_2d_context_->Flush(pp::CompletionCallback(&FlushCallback, this));
}
```

Life Stage 1: Hello, World!

Pepper: The Spice of Life

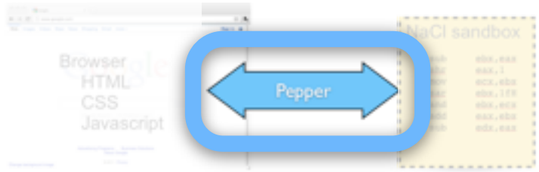



```
void Life::CreateContext(const pp::Size& size) {
  if (IsContextValid())
    return;
  graphics_2d_context_ = new pp::Graphics2D(this, size, false);
  if (!BindGraphics(*graphics_2d_context_)) {
    printf("Couldn't bind the device context\n");
  }
}

void Life::FlushPixelBuffer() {
  if (!IsContextValid())
    return;
  graphics_2d_context_->PaintImageData(*pixel_buffer_, pp::Point());
  if (flush_pending())
    return;
  set_flush_pending(true);
  graphics_2d_context_->Flush(pp::CompletionCallback(&FlushCallback, this));
}
```

Life Stage 1: Hello, World!

Pepper: The Spice of Life

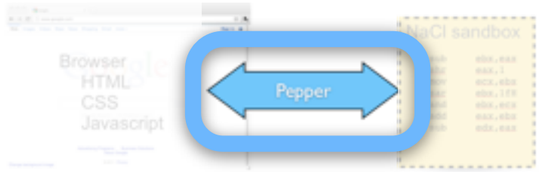



```
void Life::CreateContext(const pp::Size& size) {
  if (IsContextValid())
    return;
  graphics_2d_context_ = new pp::Graphics2D(this, size, false);
  if (!BindGraphics(*graphics_2d_context_)) {
    printf("Couldn't bind the device context\n");
  }
}

void Life::FlushPixelBuffer() {
  if (!IsContextValid())
    return;
  graphics_2d_context_->PaintImageData(*pixel_buffer_, pp::Point());
  if (flush_pending())
    return;
  set_flush_pending(true);
  graphics_2d_context_->Flush(pp::CompletionCallback(&FlushCallback, this));
}
```

Life Stage 1: Hello, World!

Pepper: The Spice of Life

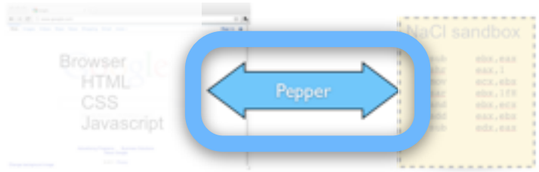



```
      printf("Couldn't bind the device context\n");
    }
  }

  void Life::FlushPixelBuffer() {
    if (!IsContextValid())
      return;
    graphics_2d_context_->PaintImageData(*pixel_buffer_, pp::Point());
    if (flush_pending())
      return;
    set_flush_pending(true);
    graphics_2d_context_->Flush(pp::CompletionCallback(&FlushCallback, this));
  }
```

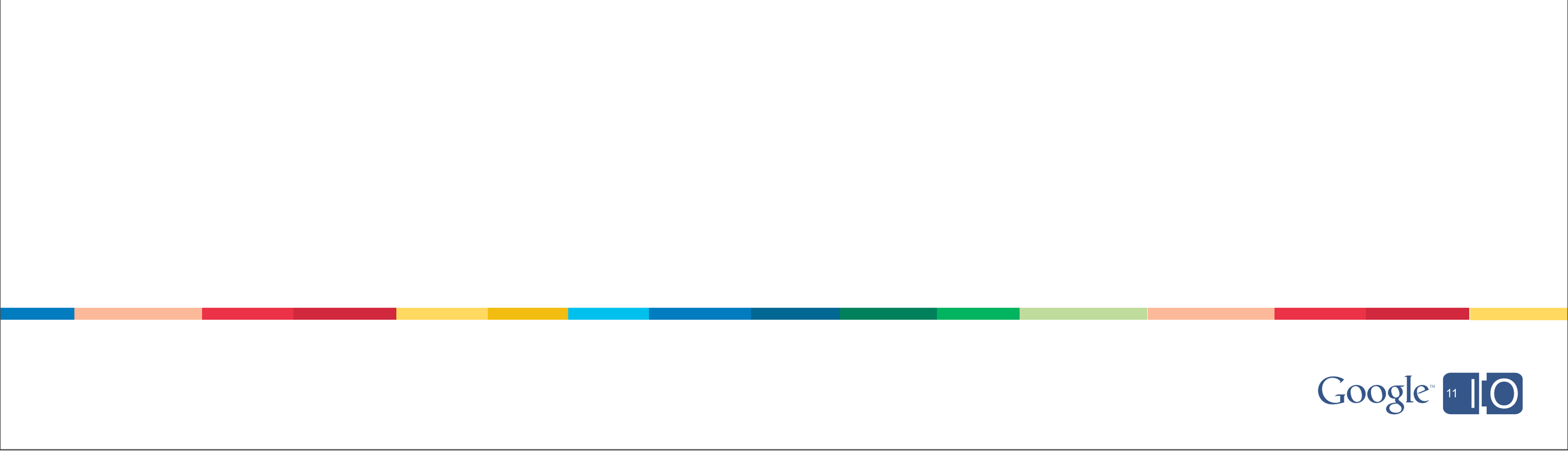
Life Stage 1: Hello, World!

Pepper: The Spice of Life




```
      printf("Couldn't bind the device context\n");
    }
  }

  void Life::FlushPixelBuffer() {
    if (!IsContextValid())
      return;
    graphics_2d_context_->PaintImageData(*pixel_buffer_, pp::Point());
    if (flush_pending())
      return;
    set_flush_pending(true);
    graphics_2d_context_->Flush(pp::CompletionCallback(&FlushCallback, this));
  }
```


Google
11

Life Stage 1: Hello, World!

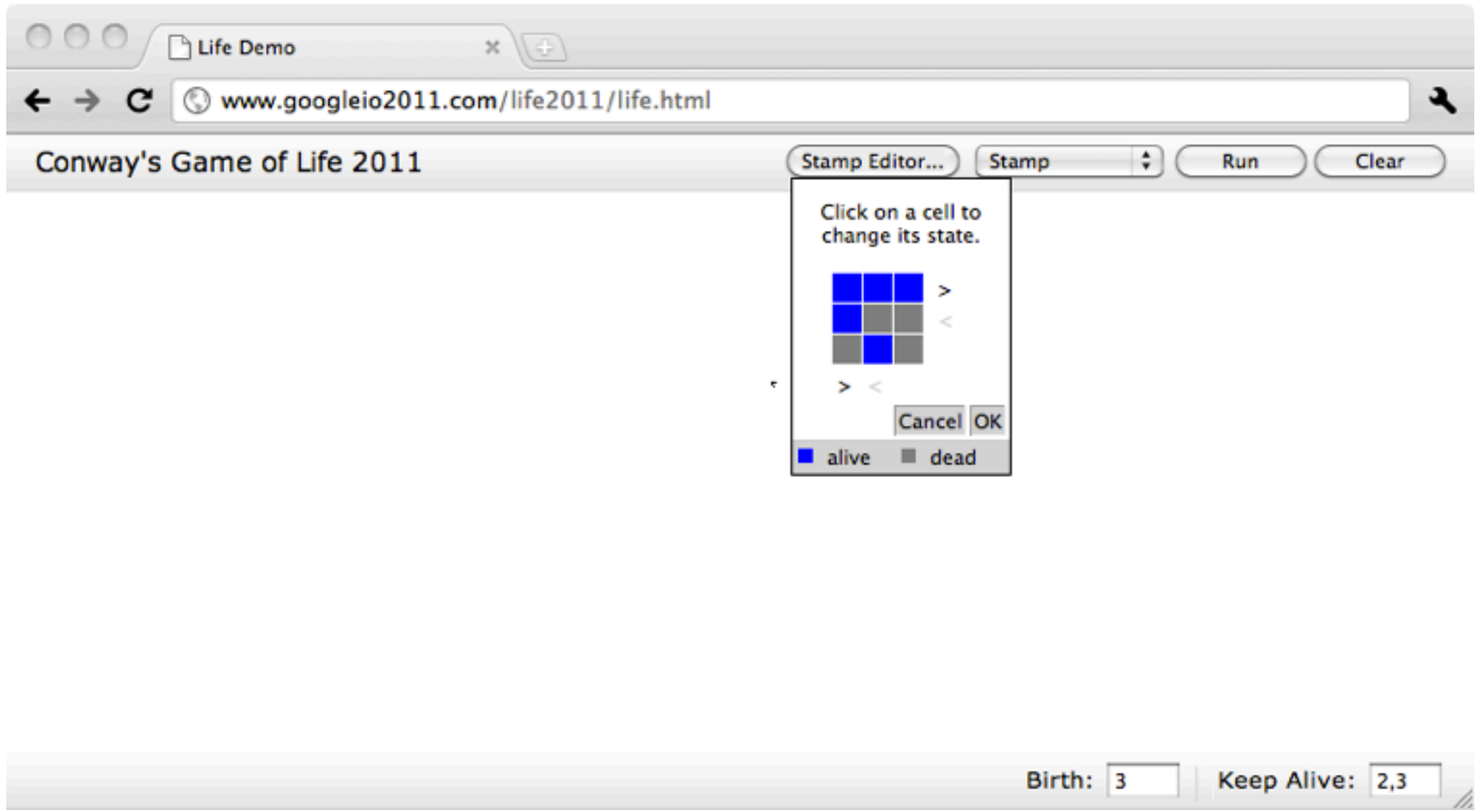
Life Stage 2: In Your Parent's House

Life in the Browser

- Make a web application
- Write the game code

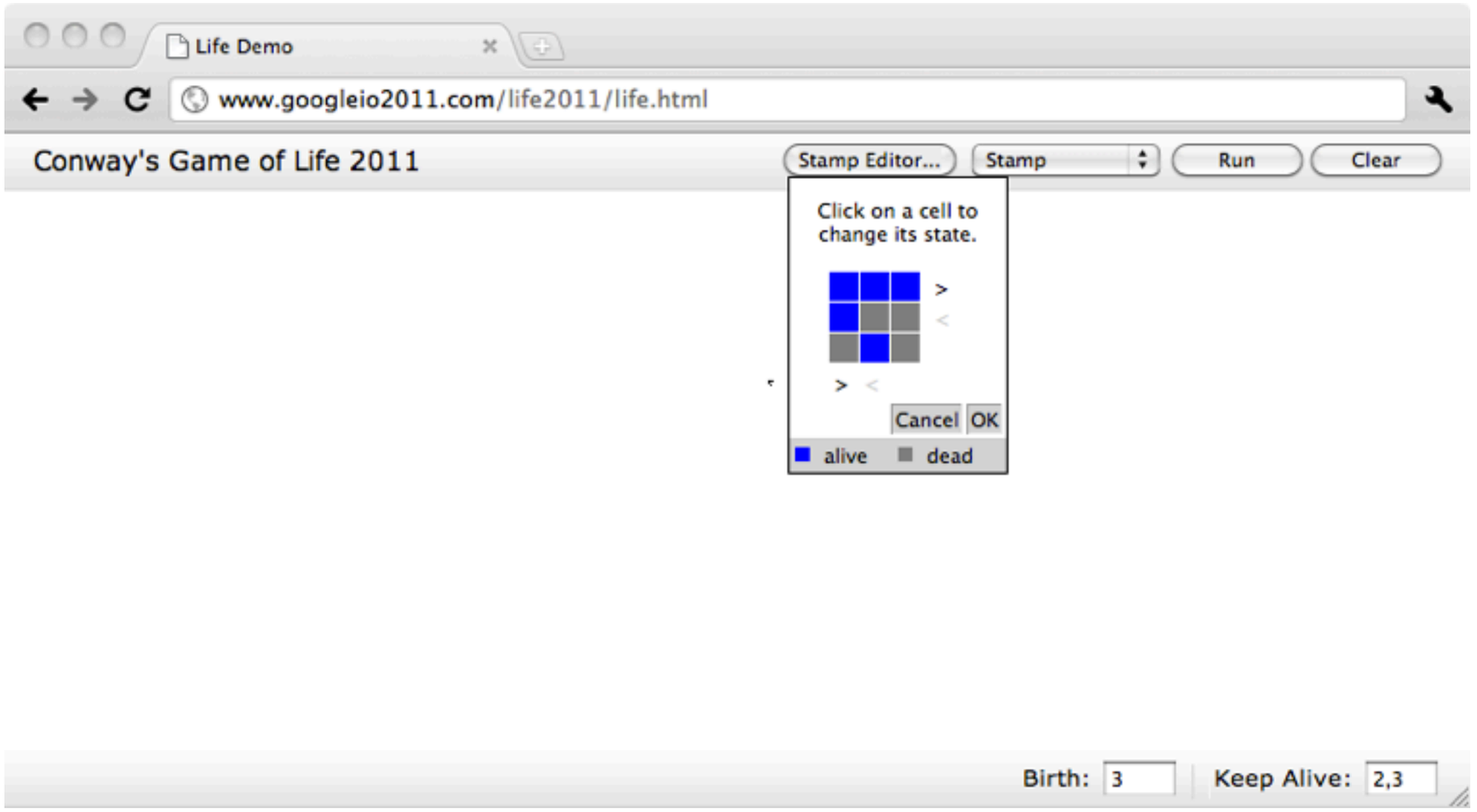
Life Stage 2: In Your Parent's House

The Stamp Collection



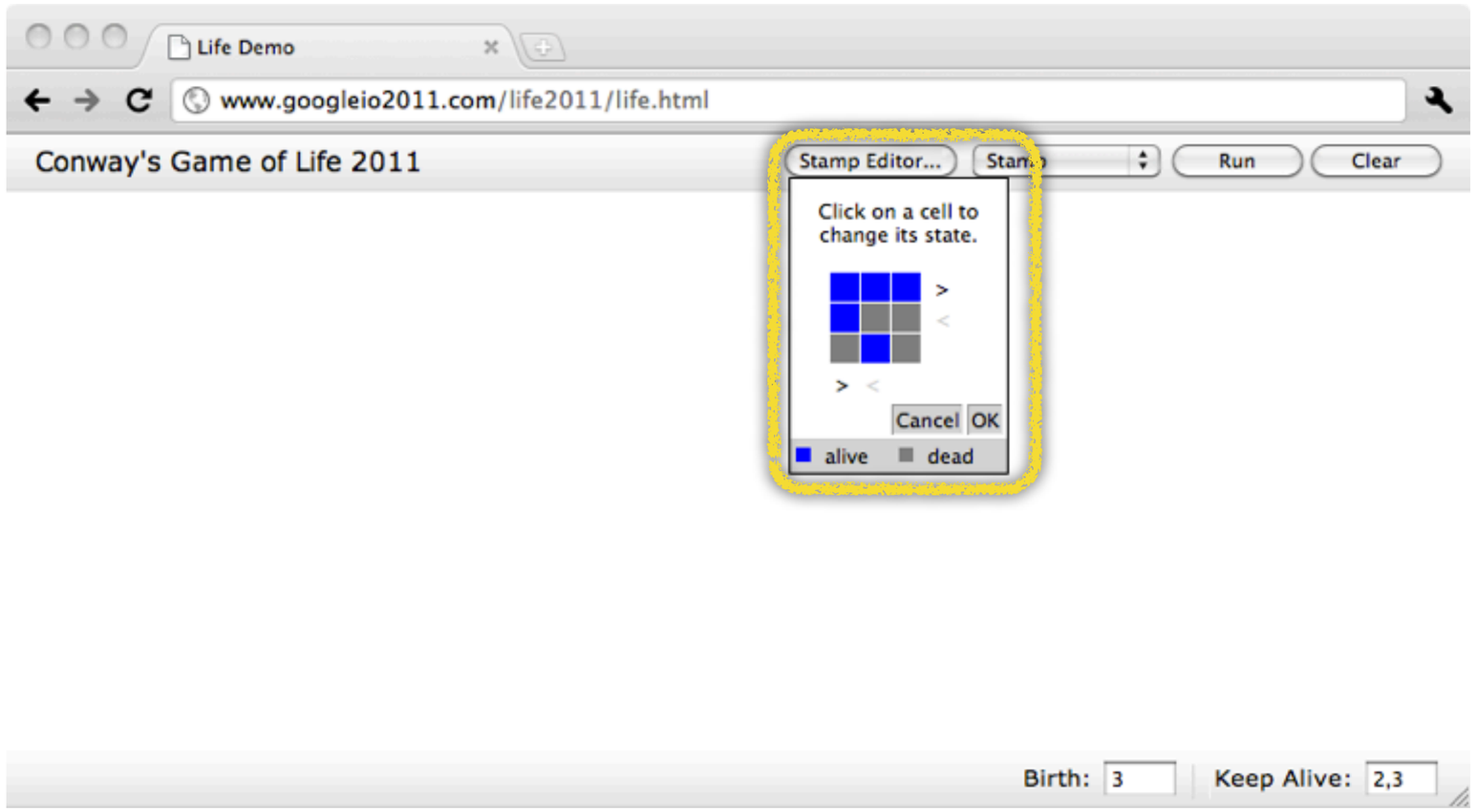
Life Stage 2: In Your Parent's House

The Stamp Collection



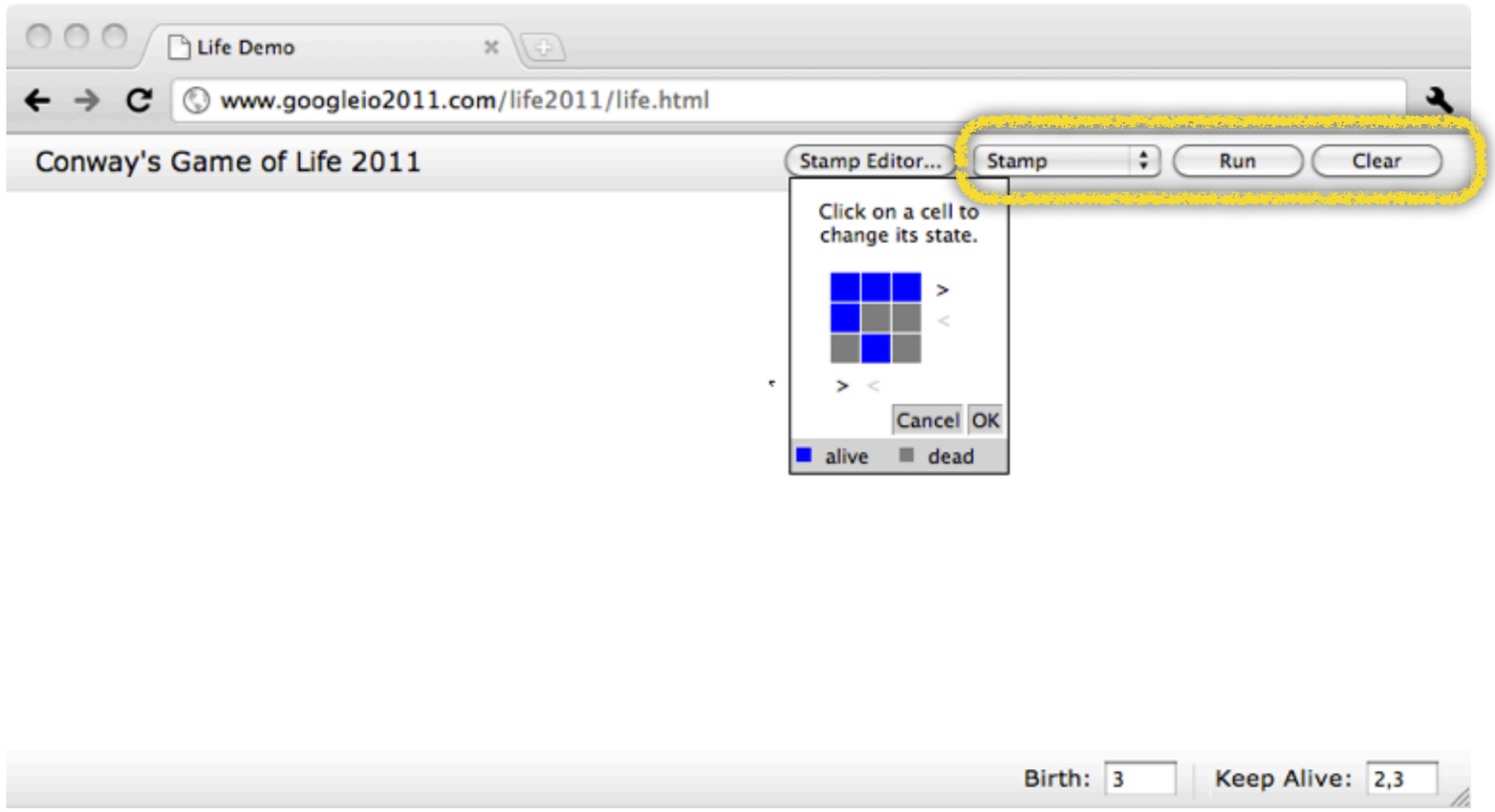
Life Stage 2: In Your Parent's House

The Stamp Collection



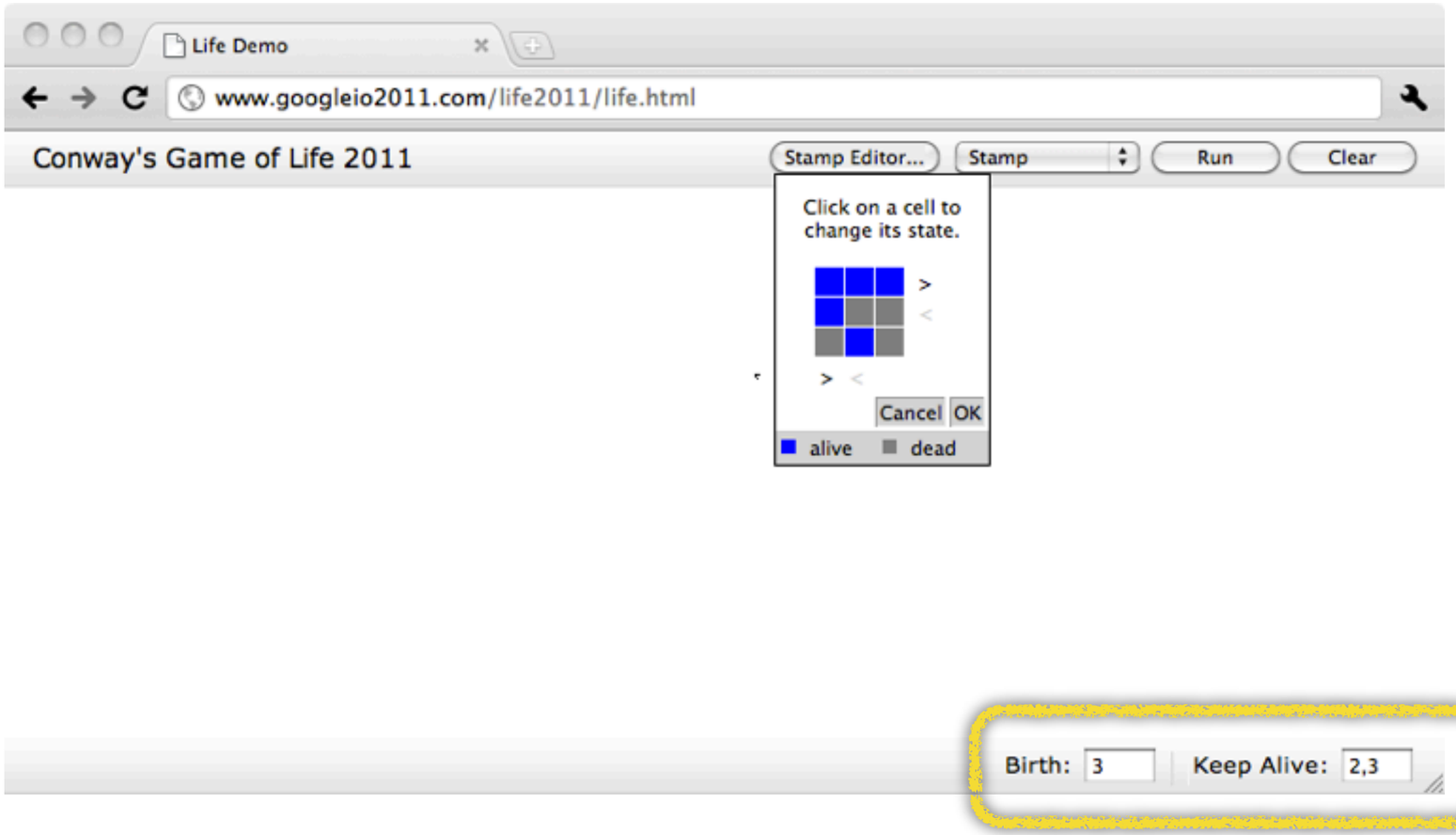
Life Stage 2: In Your Parent's House

The Stamp Collection



Life Stage 2: In Your Parent's House

The Stamp Collection



Life Stage 2: In Your Parent's House

The Stamp Collection


```
<div id="autosize_background">
  <div id="life_top_toolbar" class="life-toolbar top-toolbar">
    <div class="vertical-align-center">Conway's Game  of Life 2011</div>
    <form name="game_controls_form"
          class="vertical-align-center text-align-right"
          action=""
          method="get">
      <button type="button"
          id="stamp_editor_button"
          style="position: relative;"></button>
      <div class="panel" style="display: none;" id="stamp_editor_panel">
        <p id="title_text">Click on a cell to change its state.</p>
        <table id="stamp_editor_outer_container"
            style="font-size: inherit; text-align: center;">
          <tr>
            <!-- The cell editor table. This gets populated in stamp_editor.js -->
            <td id="stamp_editor_container" colspan=2></td>
            <td valign="top">
              <!-- The add/collapse column buttons. -->
              <table rows=2 cols=1 style="font-size: inherit;">
```

Life Stage 2: In Your Parent's House

The Stamp Collection


```
stamp.StampPanel.prototype.makeStampEditorPanel =
    function(stampEditorElements) {
  if (!stampEditorElements) {
    return null;
  }
  for (elt in stampEditorElements) {
    this.domElements_[elt] = stampEditorElements[elt];
  }
  this.domElements_.panelHeader = goog.dom.createDom('div',
      {'style': 'background-color:#EEE',
       'class': 'panel-container'}, 'Stamp Editor...');
  this.domElements_.panelContainer = stampEditorElements.mainPanel;
  goog.style.setPosition(this.domElements_.panelContainer, 0,
                         goog.style.getSize(this.parent_).height);
  this.domElements_.panelContainer.style.display = 'block';
  var newEditor = goog.dom.createDom('div', null,
      this.domElements_.panelHeader, this.domElements_.panelContainer);

  this.stampEditor_ = new stamp.Editor();
  this.stampEditor_.makeStampEditorDom(stampEditorElements.editorContainer);
```

Life Stage 2: In Your Parent's House

Learn to Listen

Life Stage 2: In Your Parent's House

Learn to Listen

- Make Life interactive
  - Process drag events in the browser

Life Stage 2: In Your Parent's House

Learn to Listen

- Make Life interactive
  - Process drag events in the browser
- Connect Javascript to C++
  - Asynchronous interface: Pepper and postMessage

Life Stage 2: In Your Parent's House

Learn to Listen

- Make Life interactive
  - Process drag events in the browser
- Connect Javascript to C++
  - Asynchronous interface: Pepper and postMessage
- Example: add a "Clear" button

Life Stage 2: In Your Parent's House

Life Can Be a Drag


```
goog.provide('uikit.events.Dragger');

goog.require('goog.fx.Dragger');

uikit.events.Dragger = function(opt_element) {
  goog.fx.Dragger.call(this, opt_element || document);
};
goog.inherits(uikit.events.Dragger, goog.fx.Dragger);
```

Life Stage 2: In Your Parent's House

Life Can Be a Drag


```
this.dragListener_ = new uikit.events.Dragger(nativeModule);
goog.events.listen(this.dragListener_, goog.fx.Dragger.EventType.START,
    this.handleStartDrag_, false, this);
goog.events.listen(this.dragListener_, goog.fx.Dragger.EventType.END,
    this.handleEndDrag_, false, this);
goog.events.listen(this.dragListener_, goog.fx.Dragger.EventType.DRAG,
    this.handleDrag_, false, this);
}

life.controllers.ViewController.prototype.handleStartDrag_ =
    function(dragStartEvent) {
  dragStartEvent.stopPropagation();
  var point = this.convertPointToWindow(
      new goog.math.Coordinate(dragStartEvent.clientX,
                               dragStartEvent.clientY));
  this.invokeMethod_('putStampAtPoint', { x: point.x, y: point.y });
};
```

Life Stage 2: In Your Parent's House

Life Can Be a Drag


```
    function(dragStartEvent) {
  dragStartEvent.stopPropagation();
  var point = this.convertPointToWindow(
      new goog.math.Coordinate(dragStartEvent.clientX,
                               dragStartEvent.clientY));
  this.invokeMethod_('putStampAtPoint', { x: point.x, y: point.y });
};
```

```
life.controllers.ViewController.prototype.handleDrag_ = function(dragEvent) {
  dragEvent.stopPropagation();
  var point = this.convertPointToWindow(
      new goog.math.Coordinate(dragEvent.clientX, dragEvent.clientY));
  this.invokeMethod_('putStampAtPoint', { x: point.x, y: point.y });
};
```

Life Stage 2: In Your Parent's House

"I'll Do It Later"


```
life.controllers.ViewController.prototype.invokeMethod_ =
    function(methodName, opt_parameters) {
  var method_invocation = methodName
  if (opt_parameters) {
    for (param in opt_parameters) {
      method_invocation += ' ' + param + ':' + opt_parameters[param]
    }
  }
  this.module_.postMessage(method_invocation);
}
```

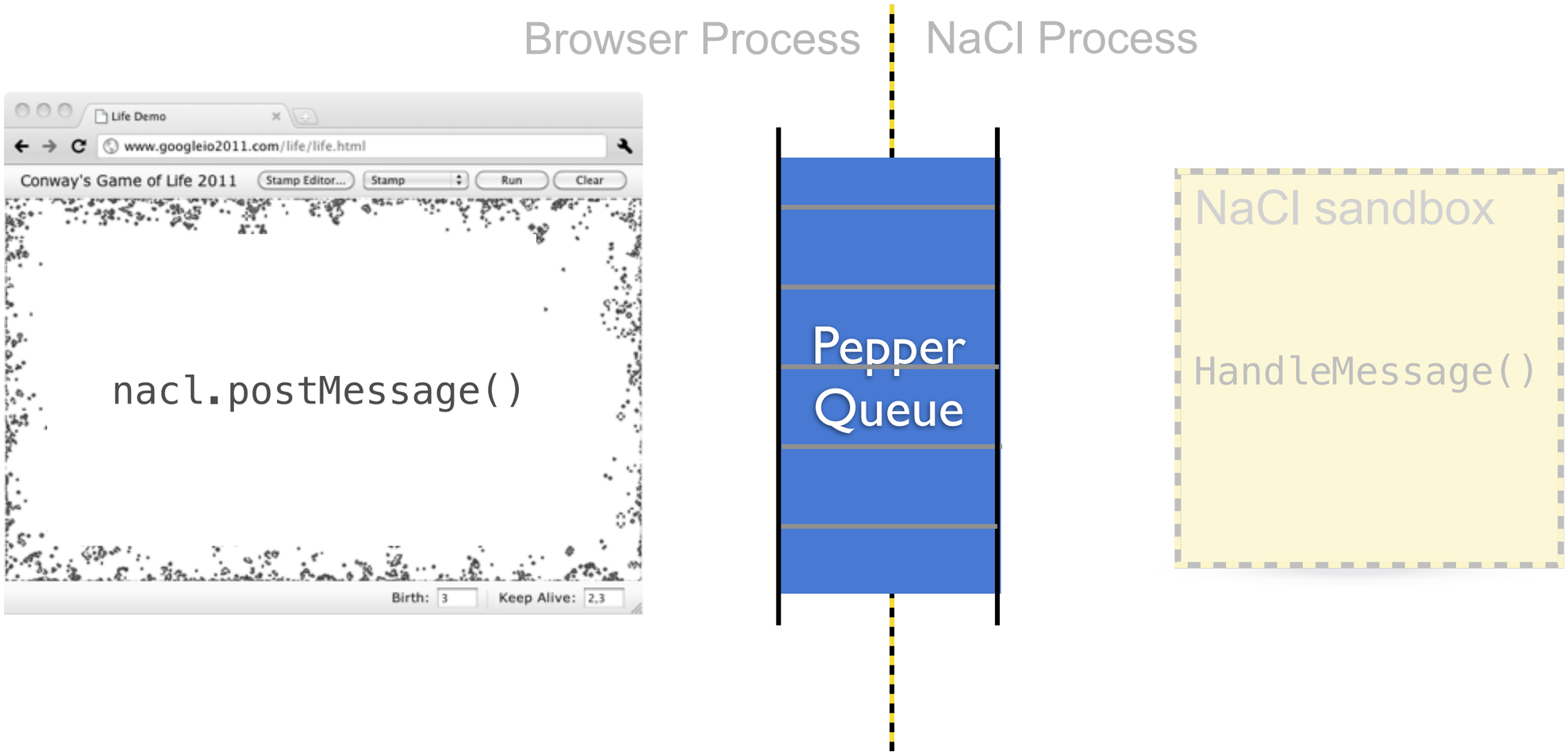
Life Stage 2: In Your Parent's House

"I'll Do It Later"


```
life.controllers.ViewController.prototype.invokeMethod_ =
    function(methodName, opt_parameters) {
  var method_invocation = methodName
  if (opt_parameters) {
    for (param in opt_parameters) {
      method_invocation += ' ' + param + ':' + opt_parameters[param]
    }
  }
  this.module_.postMessage(method_invocation);
}
```

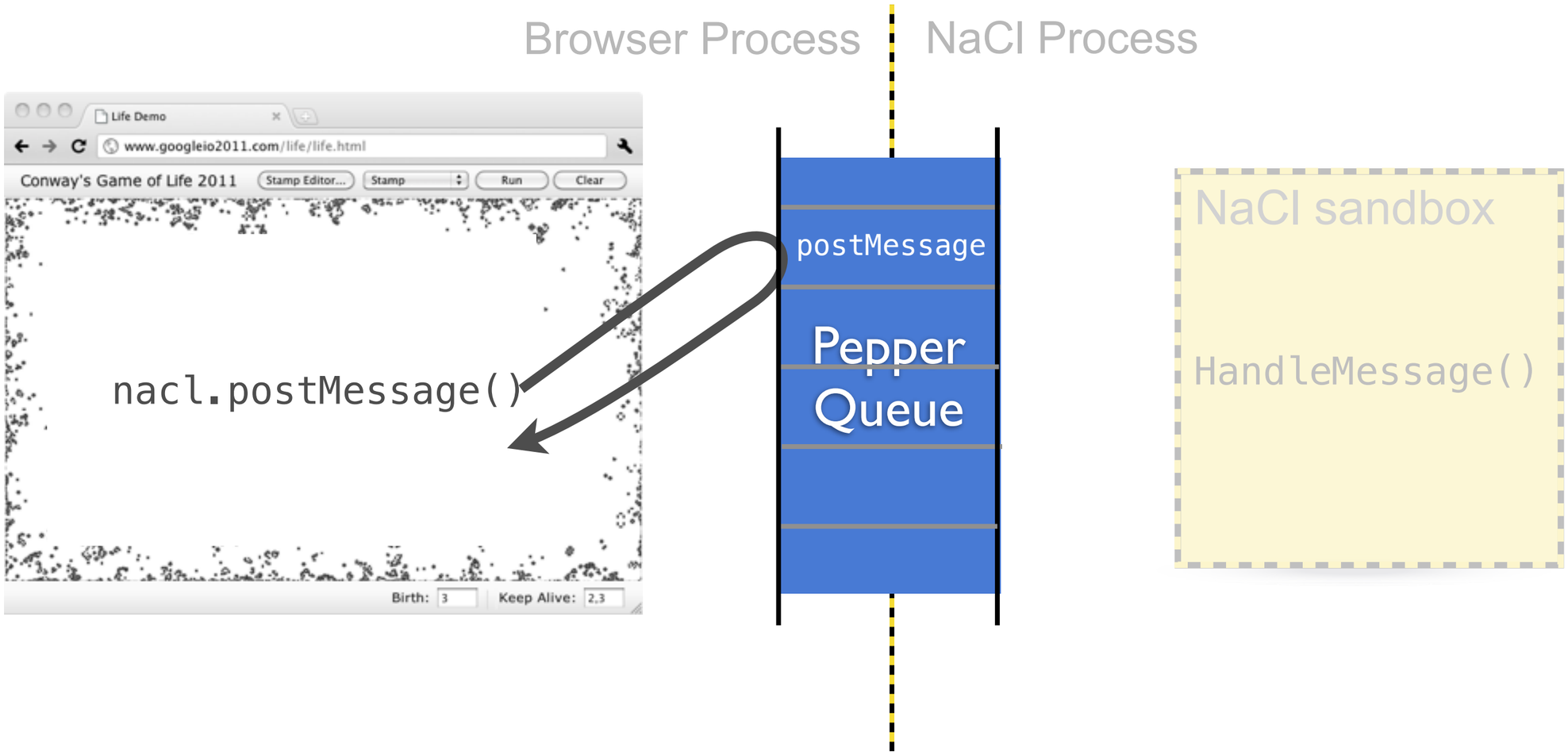
Life Stage 2: In Your Parent's House

The postMessage Trip



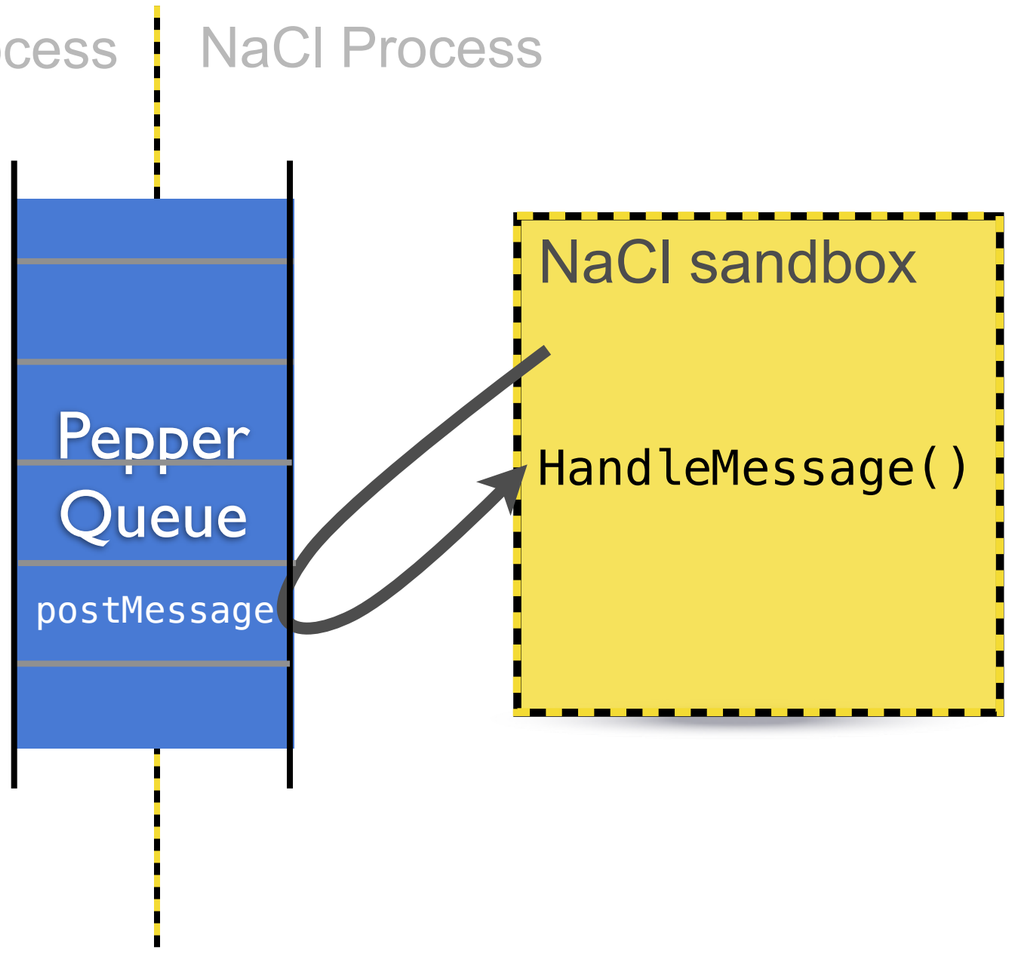
Life Stage 2: In Your Parent's House

The postMessage Trip



Life Stage 2: In Your Parent's House

The postMessage Trip



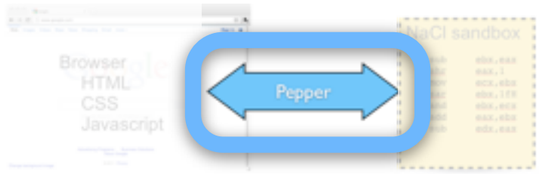
Life Stage 2: In Your Parent's House

"I'll Do It Later"


```
void LifeApplication::HandleMessage(const pp::Var& message) {
  if (!message.is_string())
    return;
  scripting_bridge_.InvokeMethod(message.AsString());
}
```

Life Stage 2: In Your Parent's House

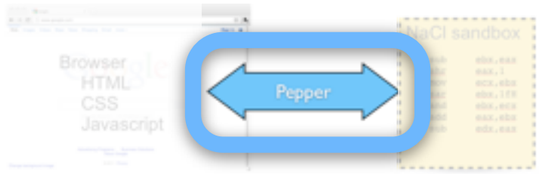
"I'll Do It Later"




```
void LifeApplication::HandleMessage(const pp::Var& message) {
  if (!message.is_string())
    return;
  scripting_bridge_.InvokeMethod(message.AsString());
}
```

Life Stage 2: In Your Parent's House

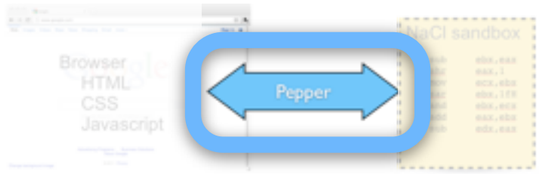
"I'll Do It Later"




```
void LifeApplication::HandleMessage(const pp::Var& message) {
  if (!message.is_string())
    return;
  scripting_bridge_.InvokeMethod(message.AsString());
}
```

Life Stage 2: In Your Parent's House

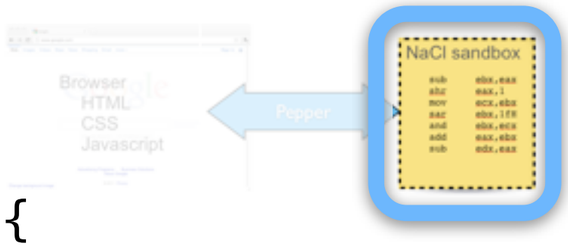
"I'll Do It Later"




```
void LifeApplication::HandleMessage(const pp::Var& message) {
  if (!message.is_string())
    return;
  scripting_bridge_.InvokeMethod(message.AsString());
}
```

Life Stage 2: In Your Parent's House

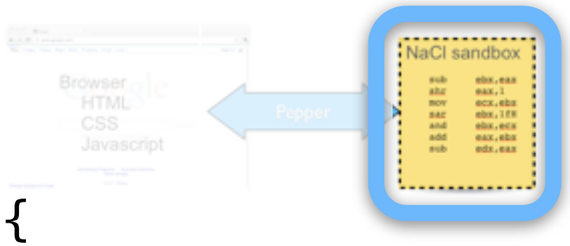
"I'll Do It Later"




```
bool ScriptingBridge::InvokeMethod(const std::string& method) {
  size_t current_pos = 0;
  const std::string method_name = ScanToken(method, &current_pos);
  MethodDictionary::iterator method_iter;
  method_iter = method_dictionary_.find(method_name);
  if (method_iter != method_dictionary_.end()) {
    std::map<std::string, std::string> param_dict;
    while (current_pos != std::string::npos) {
      const std::string parameter = ScanToken(method, &current_pos);
      if (parameter.length()) {
        std::string param_name;
        std::string param_value;
        if (ParseParameter(parameter, &param_name, &param_value)) {
          param_dict[param_name] = param_value;
        }
      }
    }
    (*method_iter->second).Execute(*this, param_dict);
    return true;
  }
```

Life Stage 2: In Your Parent's House

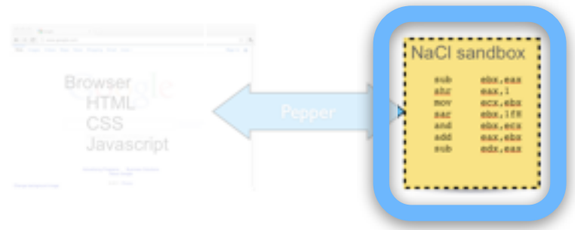
"I'll Do It Later"




```
bool ScriptingBridge::InvokeMethod(const std::string& method) {
  size_t current_pos = 0;
  const std::string method_name = ScanToken(method, &current_pos);
  MethodDictionary::iterator method_iter;
  method_iter = method_dictionary_.find(method_name);
  if (method_iter != method_dictionary_.end()) {
    std::map<std::string, std::string> param_dict;
    while (current_pos != std::string::npos) {
      const std::string parameter = ScanToken(method, &current_pos);
      if (parameter.length()) {
        std::string param_name;
        std::string param_value;
        if (ParseParameter(parameter, &param_name, &param_value)) {
          param_dict[param_name] = param_value;
        }
      }
    }
    (*method_iter->second).Execute(*this, param_dict);
    return true;
  }
```

Life Stage 2: In Your Parent's House

"I'll Do It Later"




```
        std::string param_value;
        if (ParseParameter(parameter, &param_name, &param_value)) {
          param_dict[param_name] = param_value;
        }
      }
    }
    (*method_iter->second).Execute(*this, param_dict);
    return true;
  }
  return false;
}
```

Life Stage 2: In Your Parent's House

Clear Your Mind

Life Stage 2: In Your Parent's House

Clear Your Mind

- Add the "Clear" button in HTML

Life Stage 2: In Your Parent's House

Clear Your Mind

- Add the "Clear" button in HTML
- Wire up the `onclick` event in Javascript

Life Stage 2: In Your Parent's House

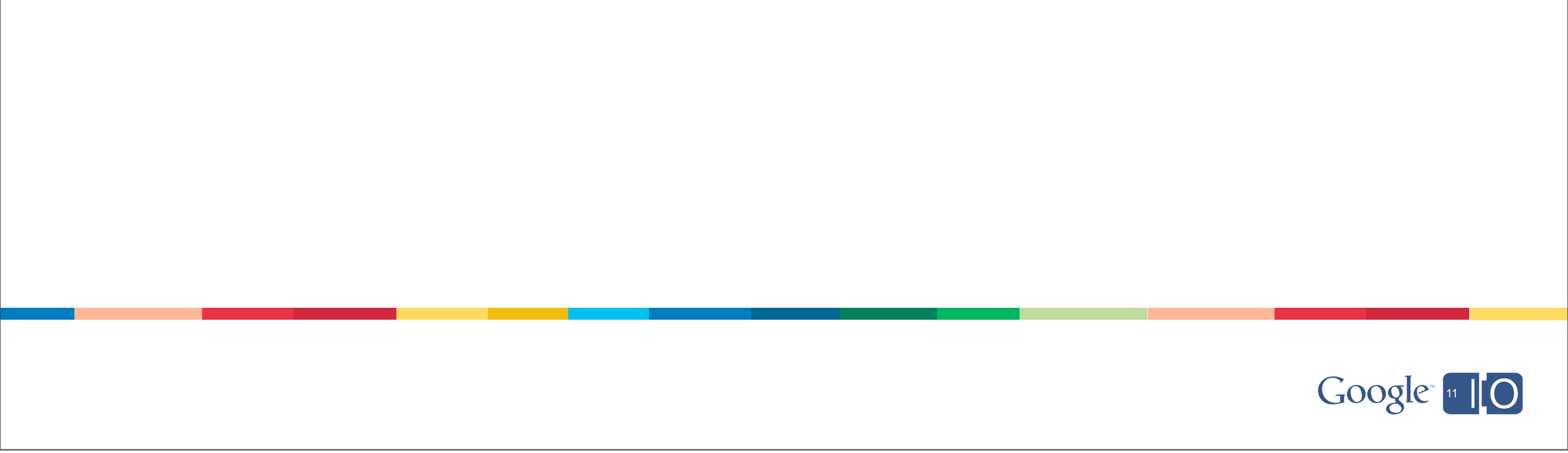
Clear Your Mind

- Add the "Clear" button in HTML
- Wire up the `onclick` event in Javascript
- Add `postMessage('clear')` to call the NaCl module

Life Stage 2: In Your Parent's House

Clear Your Mind

- Add the "Clear" button in HTML
- Wire up the `onclick` event in Javascript
- Add `postMessage('clear')` to call the NaCl module
- Add a C++ "clear" handler


Google
11

Life Stage 2: In Your Parent's House

Life Stage 3: Walk *and* Chew Gum

- Simulation on its own thread
  - Out of the browser's UI thread
  - Improve UI responsiveness

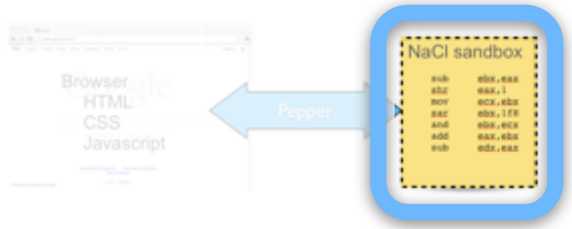
Life Stage 3: Walk *and* Chew Gum


```
void Life::StartSimulation() {
  pthread_create(&life_simulation_thread_, NULL, LifeSimulation, this);
}

Life::~Life() {
  set_is_simulation_running(false);
  if (life_simulation_thread_) {
    pthread_join(life_simulation_thread_, NULL);
  }
  DeleteCells();
  pthread_mutex_destroy(&life_simulation_mutex_);
}

Life::SimulationMode Life::WaitForRunMode() {
  simulation_mode_.LockWhenNotCondition(kPaused);
  simulation_mode_.Unlock();
  return simulation_mode();
}
```

Life Stage 3: Walk *and* Chew Gum

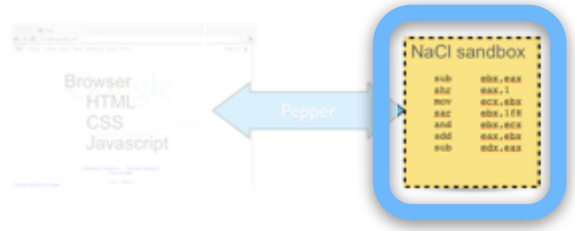



```
void Life::StartSimulation() {
  pthread_create(&life_simulation_thread_, NULL, LifeSimulation, this);
}

Life::~Life() {
  set_is_simulation_running(false);
  if (life_simulation_thread_) {
    pthread_join(life_simulation_thread_, NULL);
  }
  DeleteCells();
  pthread_mutex_destroy(&life_simulation_mutex_);
}

Life::SimulationMode Life::WaitForRunMode() {
  simulation_mode_.LockWhenNotCondition(kPaused);
  simulation_mode_.Unlock();
  return simulation_mode();
}
```

Life Stage 3: Walk *and* Chew Gum

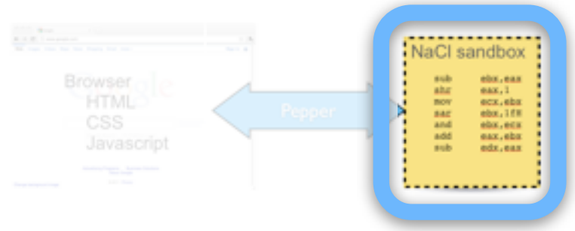



```
void Life::StartSimulation() {
  pthread_create(&life_simulation_thread_, NULL, LifeSimulation, this);
}

Life::~Life() {
  set_is_simulation_running(false);
  if (life_simulation_thread_) {
    pthread_join(life_simulation_thread_, NULL);
  }
  DeleteCells();
  pthread_mutex_destroy(&life_simulation_mutex_);
}

Life::SimulationMode Life::WaitForRunMode() {
  simulation_mode_.LockWhenNotCondition(kPaused);
  simulation_mode_.Unlock();
  return simulation_mode();
}
```

Life Stage 3: Walk *and* Chew Gum

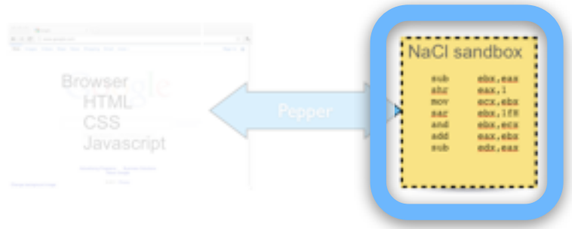



```
void Life::StartSimulation() {
  pthread_create(&life_simulation_thread_, NULL, LifeSimulation, this);
}

Life::~Life() {
  set_is_simulation_running(false);
  if (life_simulation_thread_) {
    pthread_join(life_simulation_thread_, NULL);
  }
  DeleteCells();
  pthread_mutex_destroy(&life_simulation_mutex_);
}

Life::SimulationMode Life::WaitForRunMode() {
  simulation_mode_.LockWhenNotCondition(kPaused);
  simulation_mode_.Unlock();
  return simulation_mode();
}
```

Life Stage 3: Walk *and* Chew Gum

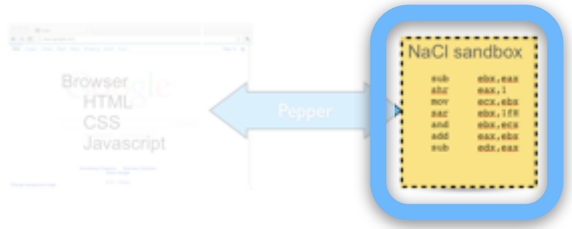



```
void Life::StartSimulation() {
  pthread_create(&life_simulation_thread_, NULL, LifeSimulation, this);
}

Life::~Life() {
  set_is_simulation_running(false);
  if (life_simulation_thread_) {
    pthread_join(life_simulation_thread_, NULL);
  }
  DeleteCells();
  pthread_mutex_destroy(&life_simulation_mutex_);
}

Life::SimulationMode Life::WaitForRunMode() {
  simulation_mode_.LockWhenNotCondition(kPaused);
  simulation_mode_.Unlock();
  return simulation_mode();
}
```

Life Stage 3: Walk *and* Chew Gum

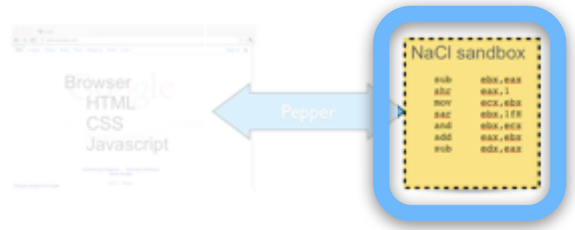



```
void Life::StartSimulation() {
  pthread_create(&life_simulation_thread_, NULL, LifeSimulation, this);
}

Life::~Life() {
  set_is_simulation_running(false);
  if (life_simulation_thread_) {
    pthread_join(life_simulation_thread_, NULL);
  }
  DeleteCells();
  pthread_mutex_destroy(&life_simulation_mutex_);
}

Life::SimulationMode Life::WaitForRunMode() {
  simulation_mode_.LockWhenNotCondition(kPaused);
  simulation_mode_.Unlock();
  return simulation_mode();
}
```

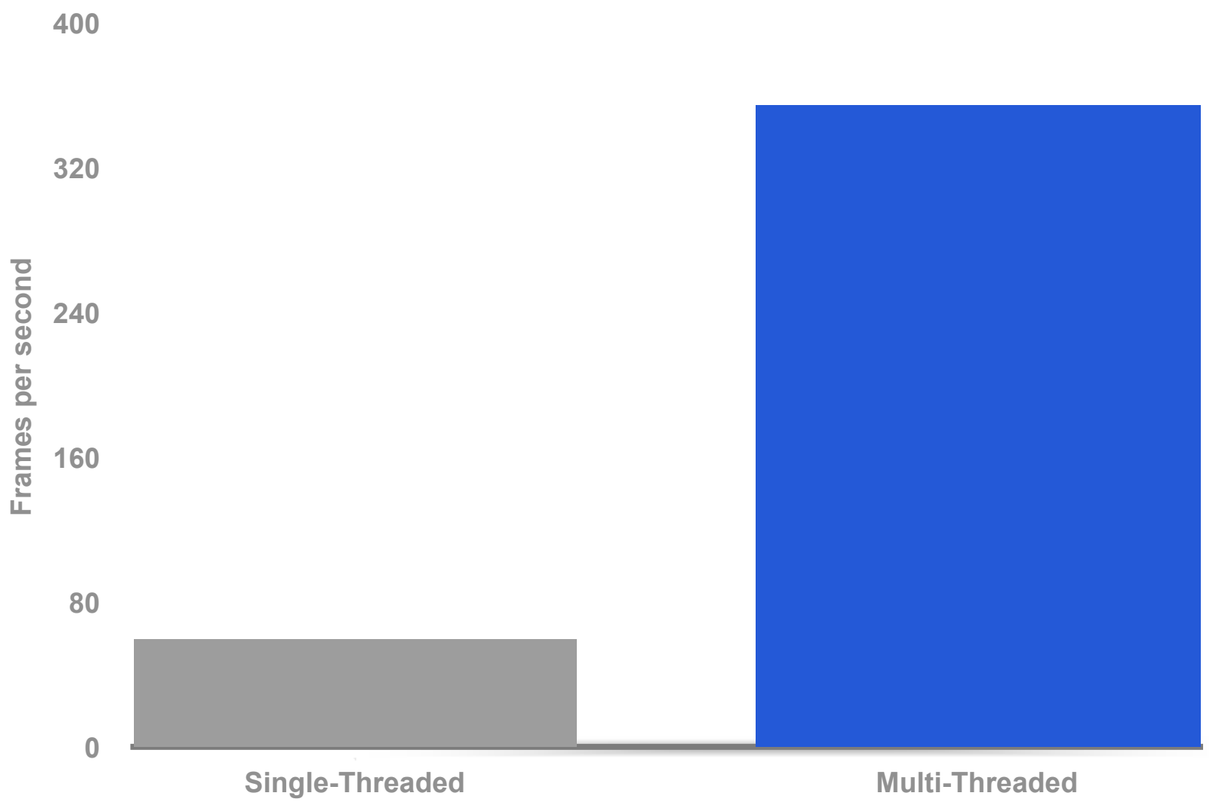
Life Stage 3: Walk *and* Chew Gum




```
void LockWhenNotCondition(int32_t condition_value) {
  Lock();
  while (condition_value == condition_value_) {
    pthread_cond_wait(&condition_condition_, &condition_lock_);
  }
  // When this method returns, |contition_lock_| will be acquired.  The
  // calling thread must unlock it.
}
void Unlock() {
  pthread_cond_broadcast(&condition_condition_);
  pthread_mutex_unlock(&condition_lock_);
}
void UnlockWithCondition(unsigned int condition_value) {
  condition_value_ = condition_value;
  Unlock();
}
```

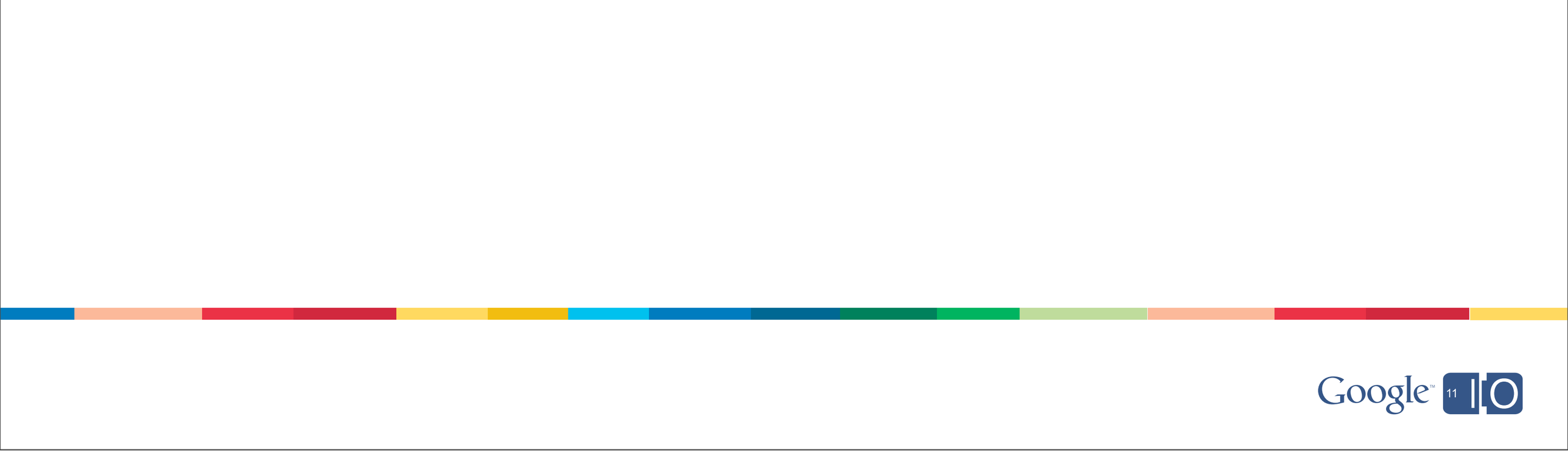
Life Stage 3: Walk *and* Chew Gum

Many Hands Make Light Work



- Decoupling threads provides about 6x speed-up
- No front-end changes required

Source: Local testing


Google
11

Life Stage 3: Walk *and* Chew Gum

Life Stage 4: Use Your Voice!

- Pepper Audio
- Pepper URL download

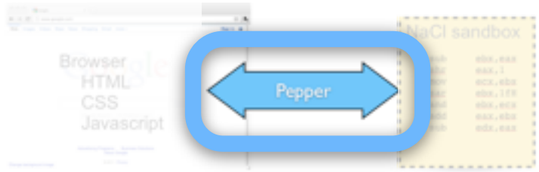
Life Stage 4: Use Your Voice

Pepper Audio


```
bool AudioPlayer::CreatePepperAudio() {
  PP_AudioSampleRate rate = (audio_source_->GetSampleRate() == 44100) ?
      PP_AUDIOSAMPLERATE_44100 : PP_AUDIOSAMPLERATE_48000;
  pp::AudioConfig config(instance_, rate, 4096);
  pp_audio_ = new pp::Audio(instance_, config, AudioCallback, this);
  return true;
}
```

Life Stage 4: Use Your Voice

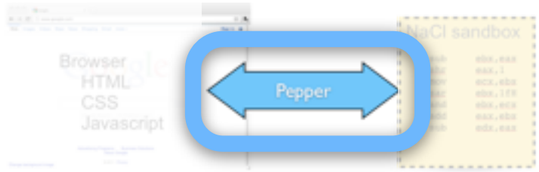
Pepper Audio




```
bool AudioPlayer::CreatePepperAudio() {
  PP_AudioSampleRate rate = (audio_source_->GetSampleRate() == 44100) ?
      PP_AUDIOSAMPLERATE_44100 : PP_AUDIOSAMPLERATE_48000;
  pp::AudioConfig config(instance_, rate, 4096);
  pp_audio_ = new pp::Audio(instance_, config, AudioCallback, this);
  return true;
}
```

Life Stage 4: Use Your Voice

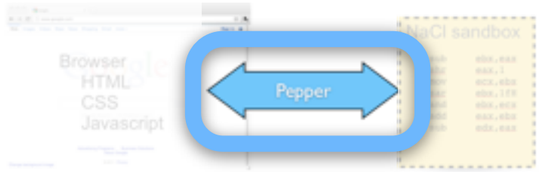
Pepper Audio




```
bool AudioPlayer::CreatePepperAudio() {
  PP_AudioSampleRate rate = (audio_source_->GetSampleRate() == 44100) ?
      PP_AUDIOSAMPLERATE_44100 : PP_AUDIOSAMPLERATE_48000;
  pp::AudioConfig config(instance_, rate, 4096);
  pp_audio_ = new pp::Audio(instance_, config, AudioCallback, this);
  return true;
}
```

Life Stage 4: Use Your Voice

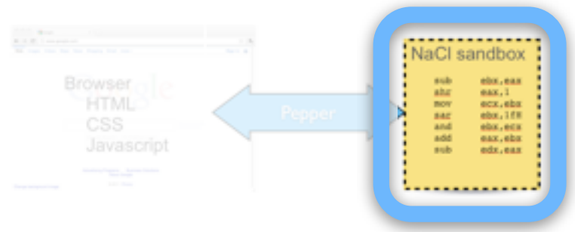
Pepper Audio




```
bool AudioPlayer::CreatePepperAudio() {
  PP_AudioSampleRate rate = (audio_source_->GetSampleRate() == 44100) ?
      PP_AUDIOSAMPLERATE_44100 : PP_AUDIOSAMPLERATE_48000;
  pp::AudioConfig config(instance_, rate, 4096);
  pp_audio_ = new pp::Audio(instance_, config, AudioCallback, this);
  return true;
}
```

Life Stage 4: Use Your Voice

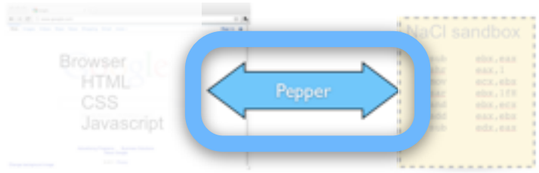
Pepper Audio




```
void AudioPlayer::AudioCallback(void* sample_buffer,
                                size_t out_buffer_size,
                                void* user_data) {
  AudioPlayer* player = reinterpret_cast<AudioPlayer*>(user_data);
  char* out_buffer = reinterpret_cast<char*>(sample_buffer);
  // Copy the next chunk of samples from the audio source.
  AudioSource* audio_source = player->audio_source_;
  const char* src_buffer = audio_source->GetAudioData();
  size_t src_buffer_size = audio_source->GetAudioDataSize();
  size_t remaining_bytes = src_buffer_size - player->playback_offset_ + 1;
  size_t copy_size = std::min(out_buffer_size, remaining_bytes);
  const char* src_buffer_start = src_buffer + player->playback_offset_;
  ::memcpy(out_buffer, src_buffer_start, copy_size);
  if (copy_size < out_buffer_size) {
    ::memset(out_buffer + copy_size, 0, out_buffer_size - copy_size);
  }
  player->playback_offset_ += copy_size;
}
```

Life Stage 4: Use Your Voice

Pepper Audio



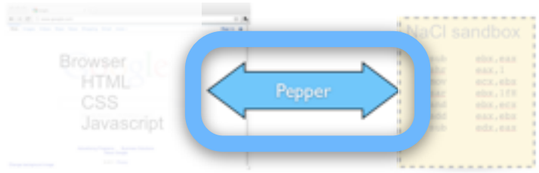

```
void AudioPlayer::Play() {
  pp::CompletionCallback cc = factory_.NewCallback(&AudioPlayer::InternalPlay);
  pp::Module::Get()->core()->CallOnMainThread(0, cc, PP_OK);
}

void AudioPlayer::InternalPlay(int32_t result) {
  assert(pp::Module::Get()->core()->IsMainThread());
  threading::ScopedMutexLock scoped_mutex(&mutex_);
  pp_audio_->StartPlayback();
}

bool AudioPlayer::CreatePepperAudio() {
  PP_AudioSampleRate rate = (audio_source_->GetSampleRate() == 44100) ?
      PP_AUDIOSAMPLERATE_44100 : PP_AUDIOSAMPLERATE_48000;
  pp::AudioConfig config(instance_, rate, 4096);
  pp_audio_ = new pp::Audio(instance_, config, AudioCallback, this);
  return true;
}
```

Life Stage 4: Use Your Voice

Pepper Audio




```
void AudioPlayer::Play() {
  pp::CompletionCallback cc = factory_.NewCallback(&AudioPlayer::InternalPlay);
  pp::Module::Get()->core()->CallOnMainThread(0, cc, PP_OK);
}

void AudioPlayer::InternalPlay(int32_t result) {
  assert(pp::Module::Get()->core()->IsMainThread());
  threading::ScopedMutexLock scoped_mutex(&mutex_);
  pp_audio_->StartPlayback();
}

bool AudioPlayer::CreatePepperAudio() {
  PP_AudioSampleRate rate = (audio_source_->GetSampleRate() == 44100) ?
      PP_AUDIOSAMPLERATE_44100 : PP_AUDIOSAMPLERATE_48000;
  pp::AudioConfig config(instance_, rate, 4096);
  pp_audio_ = new pp::Audio(instance_, config, AudioCallback, this);
  return true;
}
```

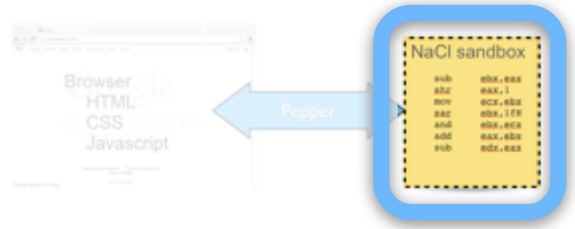
Life Stage 4: Use Your Voice

Pepper Audio


```
// Wav record descriptor per
// https://ccrma.stanford.edu/courses/422/projects/WaveFormat/
#pragma pack(push, 1)
struct WavRecord {
  // Header.
  char header_id[4];  // Should be "RIFF"
  int32_t record_size;  // Should be size of data minus 8
  char format[4];  // Should be "WAVE"
  // Subchunk 1, i.e. format descriptor.
  char format_chunk_id[4];  // Should be "fmt ".
  int32_t format_chunk_size;  // Should be 16.
  int16_t audio_format;  // Should be 1 for uncompressed audio.
  int16_t num_channels;  // 1 for mono, 2 for stereo.
  int32_t sample_rate;
  int32_t byte_rate;
  int16_t block_align;
  int16_t bits_per_sample;
  // Data chunk.
  char data_id[4];  // Should be "data"
  int32_t audio_chunk_size;  // Size of sample data that follows.
```

Life Stage 4: Use Your Voice

Pepper Audio




```
// Wav record descriptor per
// https://ccrma.stanford.edu/courses/422/projects/WaveFormat/
#pragma pack(push, 1)
struct WavRecord {
  // Header.
  char header_id[4];  // Should be "RIFF"
  int32_t record_size;  // Should be size of data minus 8
  char format[4];  // Should be "WAVE"
  // Subchunk 1, i.e. format descriptor.
  char format_chunk_id[4];  // Should be "fmt ".
  int32_t format_chunk_size;  // Should be 16.
  int16_t audio_format;  // Should be 1 for uncompressed audio.
  int16_t num_channels;  // 1 for mono, 2 for stereo.
  int32_t sample_rate;
  int32_t byte_rate;
  int16_t block_align;
  int16_t bits_per_sample;
  // Data chunk.
  char data_id[4];  // Should be "data"
  int32_t audio_chunk_size;  // Size of sample data that follows.
```

Life Stage 4: Use Your Voice

Pepper Audio


```
const WavRecord* wav = reinterpret_cast<const WavRecord*>(data);
// Check various chunk formats.
if (::strncmp(wav->header_id, "RIFF", 4) != 0 ||
    ::strncmp(wav->format, "WAVE", 4) != 0 ||
    ::strncmp(wav->format_chunk_id, "fmt ", 4) != 0 ||
    ::strncmp(wav->data_id, "data", 4) != 0) {
  printf("Did not find 'RIFF', 'WAVE' and 'fmt ' headers in wav data.\n");
  return false;
}
// Check that we support the audio format.
if (wav->sample_rate != 44100 && wav->sample_rate != 48000) {
  printf("Wav sample rate = %li. MNacl only supports 44100 or 48000.\n",
         wav->sample_rate);
  return false;
}
if (wav->num_channels != 2 || wav->bits_per_sample != 16) {
  printf("NaCl supports stereo audio and 16-bit per sample at this time.\n");
  return false;
}
// Finally, check that the actual data size and wav sample data size match.
```

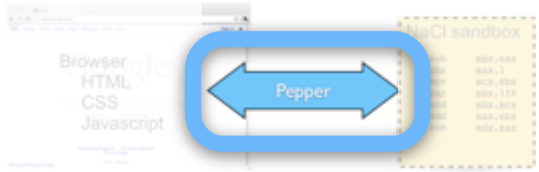
Life Stage 4: Use Your Voice

Get Webby With It


```
template <class Delegate>
void WebResourceLoader<Delegate>::StartDownload(int32_t result,
                                                const std::string& url) {
  pp::CompletionCallback cc = MakeCallback(kUrlResponseInfoReady);
  pp::URLRequestInfo request(instance_);
  InitializeRequest(url, &request);
  int32_t rv = url_loader_.Open(request, cc);
  if (rv != PP_ERROR_WOULDBLOCK) {
    cc.Run(rv);
  }
}
```

Life Stage 4: Use Your Voice

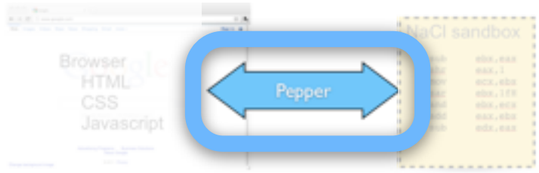
Get Webby With It




```
template <class Delegate>
void WebResourceLoader<Delegate>::StartDownload(int32_t result,
                                                const std::string& url) {
  pp::CompletionCallback cc = MakeCallback(kUrlResponseInfoReady);
  pp::URLRequestInfo request(instance_);
  InitializeRequest(url, &request);
  int32_t rv = url_loader_.Open(request, cc);
  if (rv != PP_ERROR_WOULDBLOCK) {
    cc.Run(rv);
  }
}
```

Life Stage 4: Use Your Voice

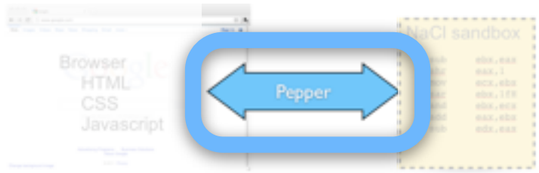
Get Webby With It




```
template <class Delegate>
void WebResourceLoader<Delegate>::StartDownload(int32_t result,
                                                const std::string& url) {
  pp::CompletionCallback cc = MakeCallback(kUrlResponseInfoReady);
  pp::URLRequestInfo request(instance_);
  InitializeRequest(url, &request);
  int32_t rv = url_loader_.Open(request, cc);
  if (rv != PP_ERROR_WOULDBLOCK) {
    cc.Run(rv);
  }
}
```

Life Stage 4: Use Your Voice

Get Webby With It

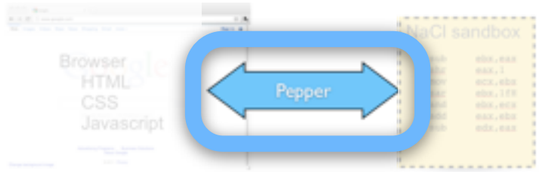



```
template <class Delegate>
void WebResourceLoader<Delegate>::InitializeRequest(
    const std::string& url,
    pp::URLRequestInfo* request) {
  request->SetURL(url);
  request->SetMethod("GET");
  request->SetFollowRedirects(true);
  request->SetStreamToFile(false);
}

template <class Delegate>
void WebResourceLoader<Delegate>::ReadNextDataBlock() {
  pp::CompletionCallback cc = MakeCallback(kDataReceived);
  int32_t rv = url_loader_.ReadResponseBody(buffer_, buffer_size_, cc);
  if (rv != PP_ERROR_WOULDBLOCK) {
    cc.Run(rv);
  }
}
```

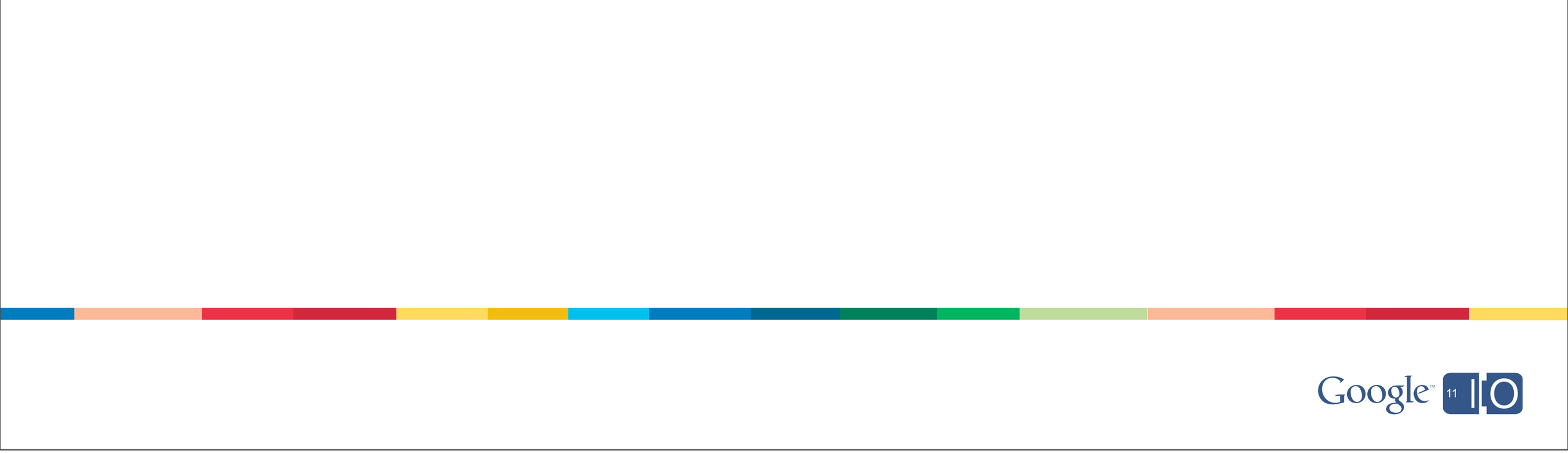
Life Stage 4: Use Your Voice

Get Webby With It




```
template <class Delegate>
void WebResourceLoader<Delegate>::ReadNextDataBlock() {
  pp::CompletionCallback cc = MakeCallback(kDataReceived);
  int32_t rv = url_loader_.ReadResponseBody(buffer_, buffer_size_, cc);
  if (rv != PP_ERROR_WOULDBLOCK) {
    cc.Run(rv);
  }
}

template <class Delegate>
void WebResourceLoader<Delegate>::StartDownload(int32_t result,
                                                const std::string& url) {
  pp::CompletionCallback cc = MakeCallback(kUrlResponseInfoReady);
  pp::URLRequestInfo request(instance_);
  InitializeRequest(url, &request);
  int32_t rv = url_loader_.Open(request, cc);
  if (rv != PP_ERROR_WOULDBLOCK) {
    cc.Run(rv);
  }
```


Google
11

Life Stage 4: Use Your Voice!

Coming Attractions

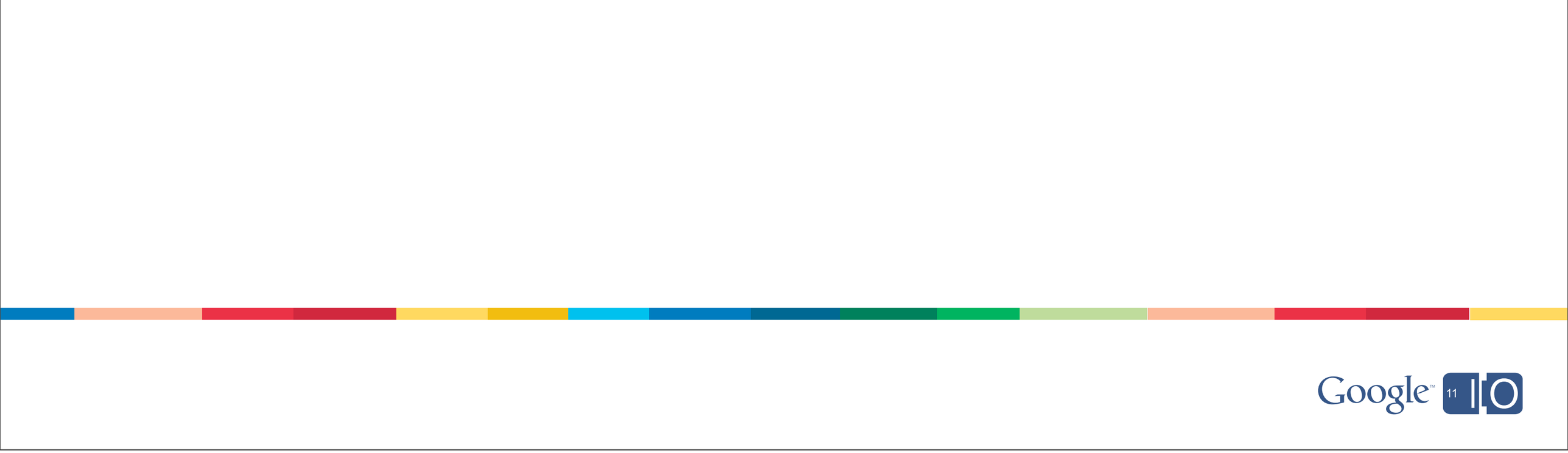
- Debugger and integrated IDE
- Garbage collection
- JIT support
- Dynamic libraries
- PNaCl
- More HTML5 support
  - 3D OpenGL
  - File I/O
  - P2P networking

Attributions

- Basic algorithm:
  - Conway's Game of Life (http://en.wikipedia.org/wiki/Conway's_Game_of_Life)
- Graphics and UI design:
  - Greg Wirt (gregw@google.com)
- Sound files:
  - WAV format reader from CCRMA (https://ccrma.stanford.edu/courses/422/projects/WaveFormat/)
  - Doorbell by Mike Koenig (http://soundbible.com/165-Door-Bell.html)
  - Other sounds from WavSource (http://www.wavsource.com/sfx/sfx.htm)

Where To Go From Here

- Documentation:
  - http://code.google.com/chrome/nativeclient/
- Life 2011 sample code:
  - http://code.google.com/p/naclports/source/browse/trunk/src/experimental/life2011/
- Visit Native Client in the Chrome Sandbox!
- Feedback:
  - http://goo.gl/sYmeZ


Google
11

Q&A


Google™
11
I/O