



ITOOOLABS 

# Embedding JavaScript into Go

GopherCon Russia 2018

Alexey Naidyonov

Mar 17 2018

- White-label cloud PBX
- IToolabs cloud or telco's on-premises
- Branded UIs & B/OSS integration
- 80+ telcos onboard
- 15 000+ SMBs total
- 1 500+ (and growing) new subscribers monthly
- 300+ mln. minutes of calls total in 2017



# What's the fuss?

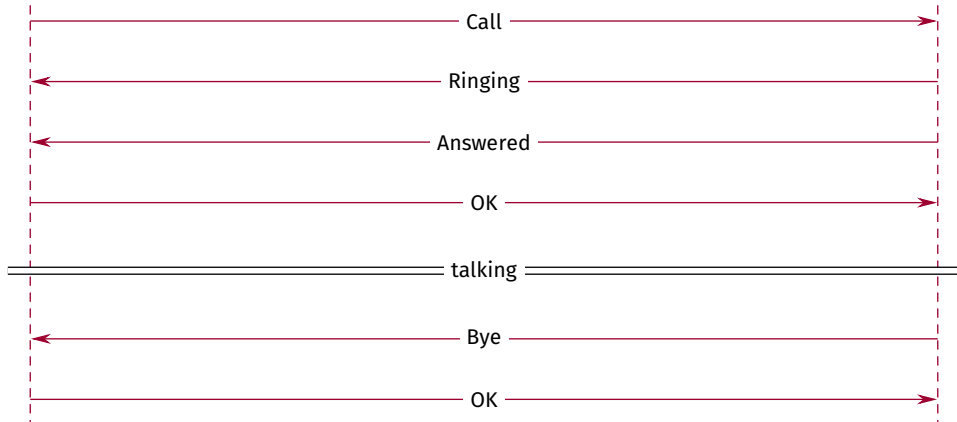
# Call model



Alice



Bob



# Call model

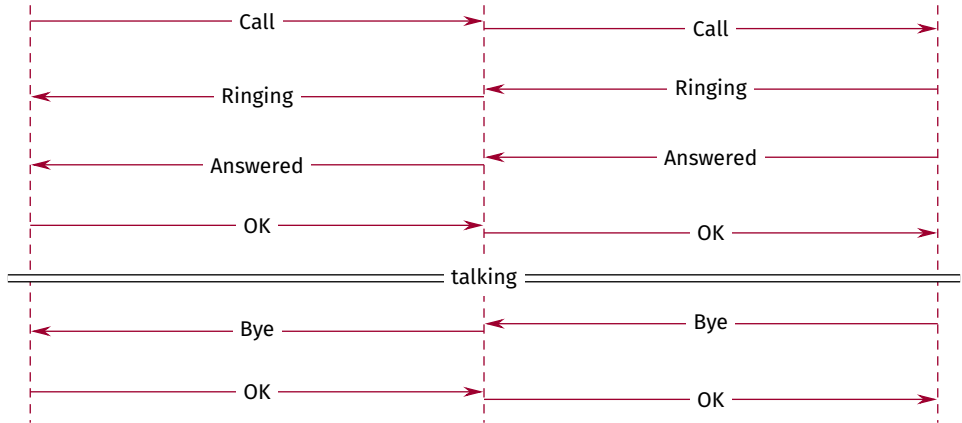


Alice

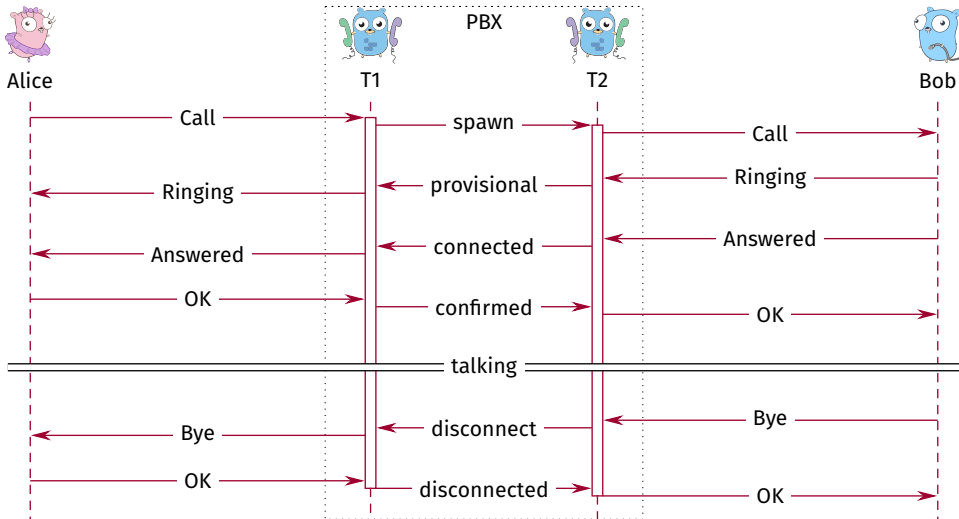
PBX



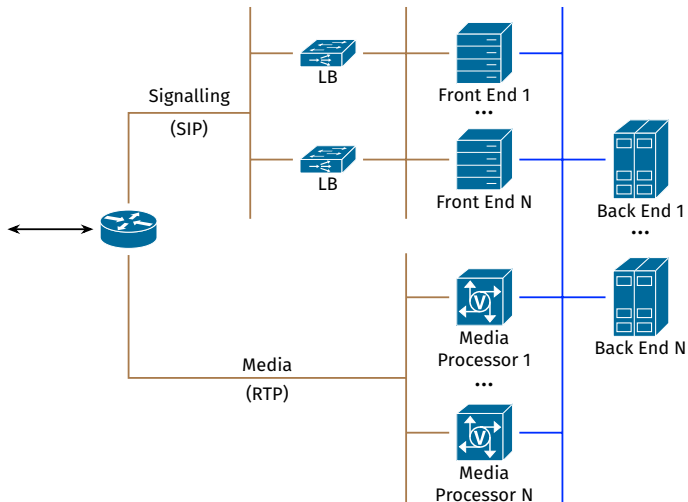
Bob



# Call model



- Easy to develop
- Easy to deploy
- Easy to scale







```
1 func Greet(runtime app.TaskRuntime, evtChan ←chan *app.TaskEvent, params
  ↪ ... interface{}) interface{} {
2   r := runtime.(app.SignalTaskRuntime)
3   if err := r.AcceptCall(); err ≠ nil {
4       return err
5   }
6   if err := r.PlayFile("http://media/greeting.mp3", false); err ≠ nil {
7       return err
8   }
9   for r.IsConnected() {
10      evt := ←evtChan
11      if r.IsPlayFileCompletedEvent(evt) {
12          break
13      }
14  }
15  return nil
16 }
```

Spawn( 'task', param) spawns new task and returns tid

ParentTask() returns tid of the parent task

ThisTask() returns tid of the current task

SendEvent(tid, what, param) sends what with param to tid

ReadInput() reads first event from the task's queue

# First try: embedding Lua 5.1 C VM

# Lua Call Processing Task



```
1 function Main()  
2     local err = AcceptCall()  
3     if err ≈ nil then  
4         return  
5     end  
6     err = PlayFile("http://media/greeting.mp3")  
7     if err ≈ nil then  
8         Disconnect()  
9         return  
10    end  
11    while IsConnected() do  
12        local evt = ReadInput(3600)  
13        if evt ≈ nil and IsPlayFileCompletedEvent(evt) then  
14            break  
15        end  
16    end  
17 end
```

# Issues



- CGo calls were way too slow (and still are)

# Issues

- CGo calls were way too slow (and still are)
- Too much hassle with conversions



- CGo calls were way too slow (and still are)
- Too much hassle with conversions
- Incomprehensible stack traces for non-terminal calls



# Second try: embedding Otto

JavaScript interpreter in Go  
([github.com/robertkrimen/otto](https://github.com/robertkrimen/otto))



# ECMAScript Call Processing Task



```
1 function Main() {  
2     AcceptCall();  
3     PlayFile('http://media/greeting.mp3');  
4     while (IsConnected()) {  
5         var evt = ReadInput(3600);  
6         if (IsDisconnectEvent(evt)) {  
7             break;  
8         }  
9     }  
10 }
```

# Issues

- Gosh, it's sooooo SLOW (100x times)

# Issues

- Gosh, it's sooooo SLOW (100x times)
- A huge burden on GC

# Issues

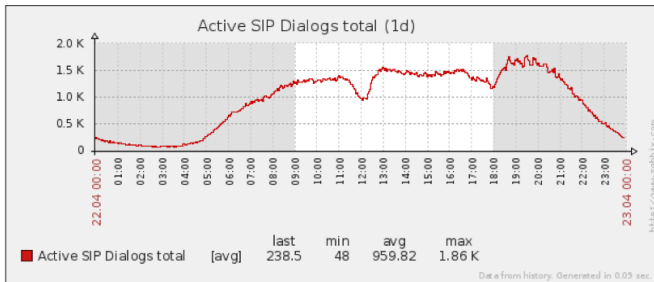
- Gosh, it's sooooo SLOW (100x times)
- A huge burden on GC

...yet it works!

# Issues

- Gosh, it's soooo SLOW (100x times)
- A huge burden on GC

...yet it works!





# Third try: rewriting Otto

Meet Goja! ([github.com/dop251/goja](https://github.com/dop251/goja))

- byte code VM



# Third try: rewriting Otto

Meet Goja! ([github.com/dop251/goja](https://github.com/dop251/goja))

- byte code VM
- lots of improvements



# Third try: rewriting Otto

Meet Goja! ([github.com/dop251/goja](https://github.com/dop251/goja))

- byte code VM
- lots of improvements
- 10x-30x times faster than Otto  
(still slow for computation intensive tasks, but pretty much fine for us)



# ECMAScript Call Processing Tasks



```
1 var call = require('lib/call.js');
2 function Main() {
3     var dest = PendingRequestData('To');
4     ProvisionCall();
5     StartBridge(Spawn('egress', dest));
6     call.process();
7 }
8 function egress(dest) {
9     var bridge = ReadInput();
10    StartBridgedCall(dest, bridge);
11    var evt = ReadInput(60);
12    if (!IsCallCompletedEvent(evt)) {
13        RejectBridge(bridge);
14    }
15    call.process();
16 }
```

```
1 function process() {
2     while (IsConnected()) {
3         var evt = ReadInput(3600);
4         if (IsDisconnectEvent(evt)) {
5             BreakBridge();
6             break;
7         }
8         if (IsBreakBridgeEvent(evt)) {
9             Disconnect();
10            break;
11        }
12    }
13 }
14 exports.process = process;
```

# There are still issues...



- Might be surprisingly slow  
(don't try to iterate over 100k+ user accounts)

# There are still issues...



- Might be surprisingly slow  
(don't try to iterate over 100k+ user accounts)
- Sharing data is an easy way to shoot oneself in the foot  
(we had to get rid of it)

# There are still issues...



- Might be surprisingly slow  
(don't try to iterate over 100k+ user accounts)
- Sharing data is an easy way to shoot oneself in the foot  
(we had to get rid of it)
- Still huge burden on GC  
(looking forward to generational GC!)

# ...but advantages are huge



- Weekly (daily, hourly) deploys

# ...but advantages are huge



- Weekly (daily, hourly) deploys

```
installapp default git "git@git.site:app/app" 0a3124f
```

## ...but advantages are huge



- Weekly (daily, hourly) deploys

```
installapp default git "git@git.site:app/app" 0a3124f
```

- Multiple applications on the same running service

## ...but advantages are huge



- Weekly (daily, hourly) deploys

```
installapp default git "git@git.site:app/app" 0a3124f
```

- Multiple applications on the same running service
- (last but not least) Same language for UI and Call Control



# Conclusions

- Embedding would suit you as long as ...

# Conclusions

- Embedding would suit you as long as ...
  - some tasks change way too often

# Conclusions



- Embedding would suit you as long as ...
  - some tasks change way too often
  - these tasks are not computation intensive ...

- Embedding would suit you as long as ...
  - some tasks change way too often
  - these tasks are not computation intensive ...
  - ...and most of time just wait for the event

- Embedding would suit you as long as ...
  - some tasks change way too often
  - these tasks are not computation intensive ...
  - ...and most of time just wait for the event
- but beware of Garbage Collector!

# Go Scripting Landscape



- <https://github.com/robertkrimen/otto> — (A very basic) ECMAScript interpreter

# Go Scripting Landscape



- <https://github.com/robertkrimen/otto> — (A very basic) ECMAScript interpreter
- <https://github.com/dop251/goja> — ECMAScript 5.1 interpreter

# Go Scripting Landscape



- <https://github.com/robertkrimen/otto> — (A very basic) ECMAScript interpreter
- <https://github.com/dop251/goja> — ECMAScript 5.1 interpreter
- <https://github.com/Shopify/go-lua> — Shopify's Go Lua



# Go Scripting Landscape



- <https://github.com/robertkrimen/otto> — (A very basic) ECMAScript interpreter
- <https://github.com/dop251/goja> — ECMAScript 5.1 interpreter
- <https://github.com/Shopify/go-lua> — Shopify's Go Lua
- <https://github.com/yuin/gopher-lua> — Go Lua

# Go Scripting Landscape



- <https://github.com/robertkrimen/otto> — (A very basic) ECMAScript interpreter
- <https://github.com/dop251/goja> — ECMAScript 5.1 interpreter
- <https://github.com/Shopify/go-lua> — Shopify's Go Lua
- <https://github.com/yuin/gopher-lua> — Go Lua
- <https://github.com/matttn/anko> — Scriptable interpreter for Go

# Go Scripting Landscape



- <https://github.com/robertkrimen/otto> — (A very basic) ECMAScript interpreter
- <https://github.com/dop251/goja> — ECMAScript 5.1 interpreter
- <https://github.com/Shopify/go-lua> — Shopify's Go Lua
- <https://github.com/yuin/gopher-lua> — Go Lua
- <https://github.com/matttn/anko> — Scriptable interpreter for Go
- <https://neugram.io/> — A Go scripting with Go Syntax

# Go Scripting Landscape



- <https://github.com/robertkrimen/otto> — (A very basic) ECMAScript interpreter
- <https://github.com/dop251/goja> — ECMAScript 5.1 interpreter
- <https://github.com/Shopify/go-lua> — Shopify's Go Lua
- <https://github.com/yuin/gopher-lua> — Go Lua
- <https://github.com/matttn/anko> — Scriptable interpreter for Go
- <https://neugram.io/> — A Go scripting with Go Syntax
- ...Tcl, Scheme, Lisp, Forth, Ruby, ...

# Questions?



THE BEST THESIS DEFENSE IS A GOOD THESIS OFFENSE.

# Author

Alexey Naidyonov

IToolLabs CEO

<https://itoolabs.com>

[anaidyonov@itoolabs.com](mailto:anaidyonov@itoolabs.com)

[@anaidyonov](https://github.com/growler)

[@anaidyonov](https://github.com/growler)

+7 926 002 40 01

