Building applications with

ERLANG

Agner Krarup Erlang

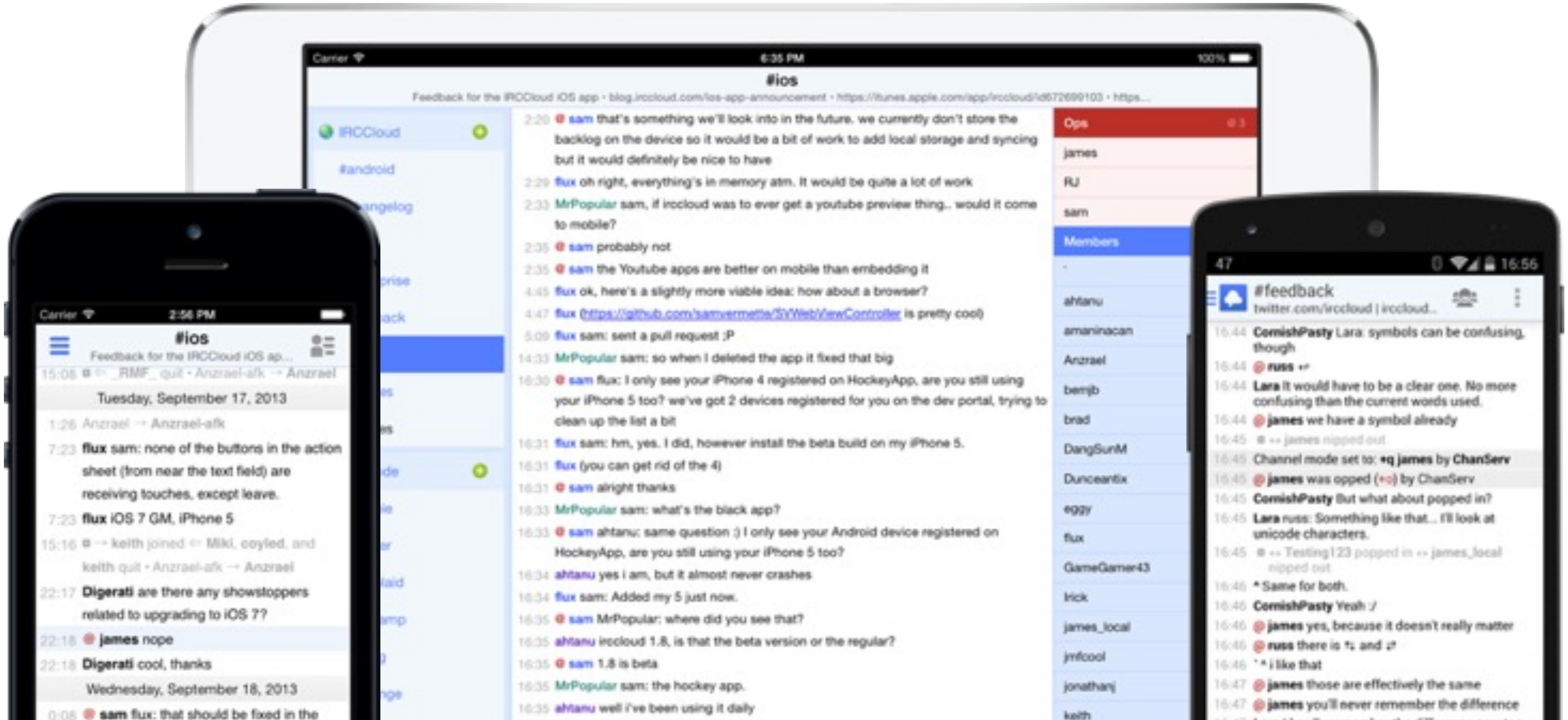
1878 - 1929

Hotline
ERICSSON
NH 51

ERICSSON
ERICSSON
GF768
YES
NO
ON OFF
CLR

IRCCloud.com

An IRC client without the baggage



Carrier
6:35 PM
100%
#ios
Feedback for the IRCCloud iOS app • blog.irccloud.com/ios-app-announcement • https://itunes.apple.com/app/irccloud/id672699103 • https...
IRCCloud
#android
Ops
Members

```
2:20 sam that's something we'll look into in the future. we currently don't store the
     backlog on the device so it would be a bit of work to add local storage and syncing
     but it would definitely be nice to have
2:29 flux oh right, everything's in memory atm. It would be quite a lot of work
2:33 MrPopular sam, if irccloud was to ever get a youtube preview thing.. would it come
     to mobile?
2:35 sam probably not
2:35 sam the Youtube apps are better on mobile than embedding it
4:45 flux ok, here's a slightly more viable idea: how about a browser?
4:47 flux (https://github.com/samvermette/SVWebViewController is pretty cool)
5:09 flux sam: sent a pull request ;P
14:33 MrPopular sam: so when I deleted the app it fixed that big
16:30 sam flux: I only see your iPhone 4 registered on HockeyApp, are you still using
      your iPhone 5 too? we've got 2 devices registered for you on the dev portal, trying to
      clean up the list a bit
16:31 flux sam: hm, yes. I did, however install the beta build on my iPhone 5.
16:31 flux (you can get rid of the 4)
16:31 sam alright thanks
16:33 MrPopular sam: what's the black app?
16:33 sam ahtanu: same question :) I only see your Android device registered on
      HockeyApp, are you still using your iPhone 5 too?
16:34 ahtanu yes i am, but it almost never crashes
16:34 flux sam: Added my 5 just now.
16:35 sam MrPopular: where did you see that?
16:35 ahtanu irccloud 1.8, is that the beta version or the regular?
16:35 sam 1.8 is beta
16:35 MrPopular sam: the hockey app.
16:35 ahtanu well i've been using it daily
```

```
Ops
james
RJ
sam
Members
ahtanu
amaninacan
Anzrael
bemjb
brad
DangSunM
Dunceantix
eggy
flux
GameGamer43
Irick
james_local
jmfcool
jonathanj
keith
```

```
15:08 ⇐ _RMF_ quit • Anzrael-afk → Anzrael
Tuesday, September 17, 2013
1:26 Anzrael → Anzrael-afk
7:23 flux sam: none of the buttons in the action
     sheet (from near the text field) are
     receiving touches, except leave.
7:23 flux iOS 7 GM, iPhone 5
15:16 → keith joined ⇐ Miki, coyled, and
      keith quit • Anzrael-afk → Anzrael
22:17 Digerati are there any showstoppers
      related to upgrading to iOS 7?
22:18 james nope
22:18 Digerati cool, thanks
Wednesday, September 18, 2013
0:08 sam flux: that should be fixed in the
```

```
16:44 CornishPasty Lara: symbols can be confusing,
      though
16:44 russ ↔
16:44 Lara It would have to be a clear one. No more
      confusing than the current words used.
16:44 james we have a symbol already
16:45 ↔ james nipped out
16:45 Channel mode set to: +q james by ChanServ
16:45 james was opped (+q) by ChanServ
16:45 CornishPasty But what about popped in?
16:45 Lara russ: Something like that... I'll look at
      unicode characters.
16:45 ↔ Testing123 popped in ↔ james_local
      nipped out
16:46 * Same for both.
16:46 CornishPasty Yeah :/
16:46 james yes, because it doesn't really matter
16:46 russ there is ⇅ and ⇄
16:46 * i like that
16:47 james those are effectively the same
16:47 james you'll never remember the difference
```

concurrent

distributed

"soft-realtime"

concurrency

light-weight process queue

Basic primitives

```
97                    % integer
1.23                  % float
hello, true, false    % atom
```

Funky primitives

```
% anonymous and named function
#Fun<mod.0.0>     fun(X) -> X+1 end
#Fun<foo.bar.1>   fun foo:bar/1

% process
<0.1.0>           spawn(Fun)  pid(0,1,0)

% port
#Port<0.0>        open_port()

% reference
#Ref<0.0.0.1>     make_ref()
```

Compound data types

```
[97,98,99]        "abc"                  % list
<<97,98,99>>  <<"abc">>                  % binary
{1,2,3}         {"a", "b", 3}            % tuple
#rec{key=value, key2="value2"}           % record
#{key => value,"key2" => "value2"}   % map

% records are just compiler sugar on tuples
{rec, value, value2}
% key names are atoms, set in their definition
-record(rec, {key, key2}).
```

```
97
1.23
hello             true   false
[97,98,99]        "abc"
<<97,98,99>>  <<"abc">>
{1,2,3}           #rec{key=value}
#{key => value}
#Fun<mod.0.0>
<0.1.0>
#Port<0.0>
#Ref<0.0.0.1>
```

hello.erl

```erlang
% usage: Pid = hello:start().

-module(hello).

-export([start/0]).

start() ->
  Pid = spawn(fun loop/0),
  % Use ! to send messages to a process
  Pid ! hello,
  Pid ! {hello, defshef16},
  Pid.
% ...
% ...
loop() ->
  receive
    hello ->
      % say hi
      io:format("Hello world!~n"),
      loop();
    {hello, Name} ->
      % say hi to someone
      io:format("Hello ~s!~n", [Name]),
      loop();
    Unrecognised ->
      % dunno
      io:format("huh? what's ~s?~n", [Unrecognised]),
      loop()
  end.
```

Building an application

OTP: Open Telecom Platform

design principles, behaviours, tooling

`gen_server`

generic server behaviour

`supervisor` trees

fault tolerant crash recovery

application

encasulation

releases

build and package with `rebar`

upgrades

hot swapping code, no downtime

demo

github.com/jwheare/defshef16

multiple function heads and guards

```
% functions with the same arity can be defined
% with multiple function heads

is_positive(X) when X > 0 ->
  true;
is_positive(X) ->
  false.
```

conditionals

`if` expressions exist, but they're weird and not very useful

```
if
  Condition ->
    do_something();
  true ->
    do_something_else()
end.

% conditions tend to be written with case expressions instead
case Condition of
  true ->
    do_something();
  false ->
    do_something_else()
end.
```

looping

loops are all about recursion, the `lists` module is our friend

```
lists:foreach(fun(X) ->
  io:format("Number: ~B!~n", [X])
end, [1,2,3]).

% sum elements in a list, [1,2,3] -> 6
Sum = lists:foldl(fun(X, Acc) ->
  Acc + X
end, 0, [1,2,3]).

% list comprehensions, increment each item
[X+1 || X <- [1,2,3]].
```

heads or tails

lists can be used as cons cells, with a head and a tail

```
List = [1,2,3]
Head = hd(List) % 1
Tail = tl(List) % [2,3]

% pipe: |, is the cons operator, used to construct lists
[Head | Tail]            % [1,2,3]
% like a russian doll
[1 | [2 | [3 | []]]] % [1,2,3]

% or for pattern matching
[First | Rest] = [1,2,3]
```