

Fractalide and Cantor

A Racket application built with Flow-Based Programming

<https://fractalide.com>

*Claes Wallin <claes.wallin@greatsinodevelopment.com>
<https://libranet.de/~clacke> (free networks -- OStatus/AP/Fediverse, Diaspora)*

Flow-Based Programming

FBP is a special case of dataflow programming characterized by asynchronous, concurrent processes "under the covers", Information Packets with defined lifetimes, named ports, "bounded buffer" connections, and definition of connections external to the components [. . .]

J Paul Rodker Morrison, <http://www.jpaulmorrison.com/fbp/>

Software reuse

Software reuse

- Modular

Software reuse

- Modular
- Object-oriented

Software reuse

- Modular
- Object-oriented
- Service-oriented

Software reuse

- Modular
- Object-oriented
- Service-oriented
- Microservices

Software reuse

- Modular
- Object-oriented
- Service-oriented
- Microservices
- Actors

Software reuse

Software reuse

- Information hiding

Software reuse

- Information hiding
- Interfaces

Software reuse

- Information hiding
- Interfaces
- Decoupling

Software reuse

- Information hiding
- Interfaces
- Decoupling
- Inversion of Control

Pipes and filters

```
$ curl -s https://web.archive.org/web/20181219004443if_/https://fractalide.com/about_us/
```

Pipes and filters

```
$ curl -s https://web.archive.org/web/20181219004443if_/https://fractalide.com/about_us/ |
```

Pipes and filters

```
$ curl -s https://web.archive.org/web/20181219004443if_/https://fractalide.com/about_us/ |  
  tidy -asxml -f /dev/null |
```

Pipes and filters

```
$ curl -s https://web.archive.org/web/20181219004443if_/https://fractalide.com/about_us/ |  
  tidy -asxml -f /dev/null |  
  xmlstarlet sel -t -m "//_:section[@id='team']//_:h2" -v . -n
```

Pipes and filters

```
$ curl -s https://web.archive.org/web/20181219004443if_/https://fractalide.com/about_us/ |  
  tidy -asxml -f /dev/null |  
  xmlstarlet sel -t -m "//_:section[@id='team']//_:h2" -v . -n
```

Stewart Mackenzie
Roy Zou
Denis Michiels
Silviu Tanase
Miguel Simoni
Claes Wallin
UX
Meredith Patterson
Kwok Ting Leung
Nixcloud

Flow-Based Programming

Coroutines meet actors meet pipes and filters

Flow-Based Programming

- Multiple inboxes (ports) and pipes (edges)

Flow-Based Programming

- Multiple inboxes (ports) and pipes (edges)
- Messages

Flow-Based Programming

- Multiple inboxes (ports) and pipes (edges)
- Messages
- Datatypes

Flow-Based Programming

- Multiple inboxes (ports) and pipes (edges)
- Messages
- Datatypes
- Feedback loops

Flow-Based Programming - Terminology

- Agent
- Port
- Edge
- Graph
- Node

Flow-Based Programming - Agent

A simple agent

```
(define-agent
 #:input '("in")
 #:output '("out")
 (define msg (recv (input "in")))
 (define option (try-recv (input "option"))))
 (if option
  (display option)
  (void))
 (displayln msg)
 (send (output "out") msg))
```

Flow-Based Programming - Graph

A semi-complex graph

```
(define-graph
  (node "vp" 'gui/vertical-panel)
  (edge-out "vp" "out" "out")
  (edge-in "in" "vp" "in")
  ...
  (node "password-label" 'gui/message)
  (edge "password-label" "out" _ "vp" "place" 60)
  (mesg "password-label" "in" '(init (label . "Password")))
  (node "password" 'cardano-wallet/password)
  (edge "password" "out" _ "vp" "place" 70)
  (mesg "password" "in" '(display . #t))
  (node "display-password" 'displayer)
  (mesg "display-password" "option" "set password: ")
  (edge "password" "password" _ "display-password" "in" _)
  ...)
```

Flow-Based Programming - Graph - LIVE



Flow-Based Programming - Graph

A semi-complex graph

```
(define-graph
  (node "vp" 'gui/vertical-panel)
  (edge-out "vp" "out" "out")
  (edge-in "in" "vp" "in")
  ...
  (node "password-label" 'gui/message)
  (edge "password-label" "out" _ "vp" "place" 60)
  (mesg "password-label" "in" '(init (label . "Password")))
  (node "password" 'cardano-wallet/password)
  (edge "password" "out" _ "vp" "place" 70)
  (mesg "password" "in" '(display . #t))
  (node "display-password" 'displayer)
  (mesg "display-password" "option" "set password: ")
  (edge "password" "password" _ "display-password" "in" _)
  ...)
```

Flow-Based Programming - App

A complete application

```
(define-graph
  (node "frame" 'gui/frame)
  (node "menu" 'cardano-wallet/menu)
  (edge "menu" "out" _ "frame" "in" _)
  (node "stack" 'gui/place-holder)
  (edge "stack" "out" _ "app" "place" 20)
  ...
  (node "display-assurance-level" 'displayer)
  (mesg "display-assurance-level" "option" "display-assurance-level: ")
  (edge "wsettings" "assurance-level" _ "display-assurance-level" "in" _)
  (mesg "wsettings" "assurance-level" "Medium")
  ...
  (node "card-display" 'plumbing/demux)
  (mesg "card-display" "option"
    (match-lambda [(cons dest _) (list (list* dest 'display #t))]))
  (edge "sidebar" "choice" _ "card-display" "in" _)
  ...)
```

Cardano wallet



Flow-Based Programming - App

A complete application

```
(define-graph
  (node "frame" 'gui/frame)
  (node "menu" 'cardano-wallet/menu)
  (edge "menu" "out" _ "frame" "in" _)
  (node "stack" 'gui/place-holder)
  (edge "stack" "out" _ "app" "place" 20)
  ...
  (node "display-assurance-level" 'displayer)
  (mesg "display-assurance-level" "option" "display-assurance-level: ")
  (edge "wsettings" "assurance-level" _ "display-assurance-level" "in" _)
  (mesg "wsettings" "assurance-level" "Medium")
  ...
  (node "card-display" 'plumbing/demux)
  (mesg "card-display" "option"
    (match-lambda [(cons dest _) (list (list* dest 'display #t))]))
  (edge "sidebar" "choice" _ "card-display" "in" _)
  ...)
```

Future

Future

Trulity

- IDE
- Browser
- NDN

Future

Mercat Cross

- Component marketplace
- App marketplace
- Solving the OpenSSL problem

Future

Luceo (CEO)

- Initial funding for Fractalide
- Enable transactions on Mercat Cross
- Ouroboros Cryptsinous (DPoS)

