INTEGRANT

the best use case for
derived keywords, I’ve seen so far

a micro framework

like component or mount

inspired by arachne

library vs. frame work

libraries

libraries

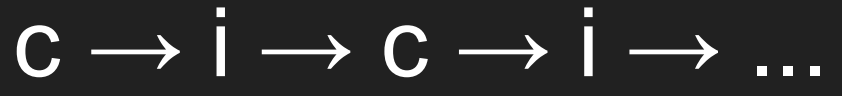
framework

```clojure
(def database
  (db/connect “jdbc:sqlite:”))

(defn handler [request]
  (resp/response (query-status database)))

(defn -main []
  (jetty/run-jetty handler {:port 8080}))
```

configuration & dependencies

dependencies are a form of configuration

configuration → implementation


c → i → c → i → ...

```
{:database.sql/connection
 {:uri “jdbc:sqlite:”}

 :example/handler
 {:database #ref :database.sql/connection}

 :ring.adapter/jetty
 {:handler #ref :example/handler
  :port 8080}}
```

symbols in namespaces
behave like
keywords in a map

```
{:database.sql/connection
 {:uri “jdbc:sqlite:”}

 :example/handler
 {:database #ref :database.sql/connection}

 :ring.adapter/jetty
 {:handler #ref :example/handler
  :port 8080}}
```

edn tagged elements

topological order

this is the trunk of the tree
minus the code

configuration → implementation ?

```
{:database.sql/connection
 {:uri “jdbc:sqlite:”}

 :example/handler
 {:database #ref :database.sql/connection}

 :ring.adapter/jetty
 {:handler #ref :example/handler
  :port 8080}}
```

```clojure
(def database
  (db/connect “jdbc:sqlite:”))

(defn handler [request]
  (resp/response (query-status database)))

(defn -main []
  (jetty/run-jetty handler {:port 8080}))
```

multimethods

```clojure
(defmethod ig/init-key :database.sql/connection
  [_ {:keys [uri]}]
  (db/connect uri))

(defmethod ig/init-key :example/handler
  [_ {:keys [database]}]
  (fn [request]
    (resp/response (query-status database))))

(defmethod ig/init-key :ring.adapter/jetty
  [_ {:keys [handler] :as options]
  (jetty/run-jetty handler options))
```

any benefits?

separation of structure from code

plurality & life cycles

reusability

```clojure
(defmethod ig/init-key :database.sql/connection
  [_ {:keys [uri]}]
  (db/connect uri))

(defmethod ig/init-key :example/handler
  [_ {:keys [database]}]
  (fn [request]
    (resp/response (query-status database))))

(defmethod ig/init-key :ring.adapter/jetty
  [_ {:keys [handler] :as options]
  (jetty/run-jetty handler options))
```

transparency

taxonomy

multiple inheritance

derived keywords

```
(derive :ring.adapter/jetty :ring/adapter)
```

modularity & extensibility

That’s it.

Thanks for your attention.

Now^H questions, please.