# Hands-on Datomic Experience

Jochen Rau

Cambridge, July 9, 2015

# Main Characteristics

- database = immutable value
- time-aware
- granular information model (Datom)
- flexible schema
- database functions
- declarative query language (Datalog)
- queries = data

# EDN Data Structure

```
What we know about :person1

{:first-name "Jochen"
 :last-name  "Rau"
 :age        44
 :address    {:locality             "Amherst"
              :administrative-area "MA"
              :post-code           "01002"
              :thoroughfare        "120 Pulpit Hill Rd."
              :premises              "Unit 23"}
 :phone      #{{:area-code      "413"
                :exchange-code "695"
                :local-number  "2315"
                :usage          [:private :primary]
                :media-type     :mobile
                :valid-from     #inst "2012-11-01T00:00:00+00:00"}}}
```

# Datoms

```
[:person1 :first-name "Jochen"]
```

entity attribute value

EAV

# Datoms (now really)

```
[:person1 :first-name "Jochen" 1234 true]
```

entity attribute value tx added

EAVT

# Datoms (now really)

```
[:person1 :first-name "Jochen" 1234 true]
```


```
[0x00002F       678    "Jochen" 1234 true]
```

# Datoms

```
[:person1 :first-name "Jochen"]
[:person1 :age 44]
[:person1 :address :addr1]
[:person1 :phone :ph1]

[:addr1 :locality "Amherst"]

[:ph1 :area-code "413"]
[:ph1 :usage :private]
[:ph1 :usage :primary]
```

scalar values

references

# Transactions

```
API
[:db/add entity-id attribute value]
[:db/retract entity-id attribute value]

1. Transaction: asserting some facts
[:db/add 42 :first-name "Jochen"]
[:db/add 42 :last-name "Rau"]
[:db/add 42 :age 45]
[:db/add 42 :likes "Veggies"]

2. Transaction: asserting some more facts
[:db/retract 42 :likes "Veggies”] explicit retraction
[:db/add 42 :likes "Pizza"]

3. Transaction: asserting some more facts
[:db/add 42 :age 45] implicit retraction
                     (there is only one :age – cardinality one)
```

# Transactions

```
Vector form
[:db/add 42 :first-name "Jochen"]
[:db/add 42 :last-name "Rau"]
[:db/add 42 :age 45]
[:db/add 42 :likes "Veggies"]

Map form
{:db/id       42          mandatory
 :first-name "Jochen"
 :last-name  "Rau"
 :age        45
 :likes      "Veggies"}

only addition, no retraction other than implicit
```

# Transactions

```
{:db/id          #db/id[:db.part/user -2]
 :ex/area-code      "413"
 :ex/exchange-code "695"
 :ex/local-number  "2315"}

{:db/id          #db/id[:db.part/user]
 :ex/first-name "Jochen"
 :ex/last-name  "Rau"
 :ex/age        44
 :ex/likes      “Veggies"
 :ex/phone      #db/id[:db.part/user -2]}
```

reference

# Transactions

```
{:db/id                  #db/id [:db.part/user -300015]
 :resource/id            #uuid "16447494-fa8c-4f6f-90d4-6a68eff9d7c8"
 :xnl/person-name        [{:xnl/person-name-type :xnl.person-name-type/LegalName
                           :xnl/name-element [{:content/value "Jochen"
                                               :xnl/element-type :xnl.element-type/FirstName}
                                              {:content/value "Rau"
                                               :xnl/element-type :xnl.element-type/LastName}]}]
 :coho/membership        [{:coho/party #db/id [:db.part/user -200023]
                           :coho/valid-from #inst "2012-11-01T00:00:00+00:00"}]
 :xpil/birth-info        {:xpil/birth-date-time     #inst "1971-03-15T00:00:00+00:00"
                          :coho/date-time-precision :coho.date-time-precision/DayMonthYear}
 :xpil/electronic-address [{:xpil/identifier "jochen@rau-family.us"
                            :xpil/service :xpil.service/Email
                            :xpil/usage :xpil.usage/Private}
                           {:xpil/identifier "jrau@infinitecloud.com"
                            :xpil/service :xpil.service/Email
                            :xpil/usage :xpil.usage/Business}
                           {:xpil/identifier "jocrau"
                            :xpil/service :xpil.service/Skype
                            :xpil/usage :xpil.usage/Private}]
 :xpil/telephone-number   [{:xpil/area-code "413"
                            :xpil/exchange-code "695"
                            :xpil/local-number "2315"
                            :xpil/usage [:xpil.usage/Private :xpil.usage/Primary]
                            :xpil/media-type :xpil.media-type/Mobile}]}
```

# Schema

```
{:db/id #db/id[:db.part/db]
 :db/ident :user/first-name
 :db/valueType :db.type/string
 :db/cardinality :db.cardinality/one
 :db.install/_attribute :db.part/db}

{:db/id #db/id[:db.part/db]
 :db/ident :user/age
 :db/valueType :db.type/long
 :db/cardinality :db.cardinality/one
 :db.install/_attribute :db.part/db}

{:db/id #db/id[:db.part/db]
 :db/ident :user/phone-number
 :db/valueType :db.type/ref
 :db/cardinality :db.cardinality/many
 :db.install/_attribute :db.part/db}
```

# Schema

```
{:db/id #db/id[:db.part/db]
 :db/ident :user/first-name
 :db/valueType :db.type/string
 :db/cardinality :db.cardinality/one
 :db.install/_attribute :db.part/db}

{:db/id #db/id[:db.part/db]
 :db/ident :user/age
 :db/valueType :db.type/long
 :db/cardinality :db.cardinality/one
 :db.install/_attribute :db.part/db}

{:db/id #db/id[:db.part/db]
 :db/ident :user/phone-number
 :db/valueType :db.type/ref
 :db/cardinality :db.cardinality/many
 :db.install/_attribute :db.part/db}
```

```
:db.type/string
:db.type/boolean
:db.type/long
:db.type/bigint
:db.type/float
:db.type/double
:db.type/bigdec
:db.type/ref
:db.type/instant
:db.type/uuid
:db.type/uri
:db.type/bytes
```

# Schema

```
{:db/id #db/id[:db.part/db]
 :db/ident :user/first-name
 :db/valueType :db.type/string
 :db/cardinality :db.cardinality/one
 :db.install/_attribute :db.part/db}

{:db/id #db/id[:db.part/db]
 :db/ident :user/age
 :db/valueType :db.type/long
 :db/cardinality :db.cardinality/one
 :db.install/_attribute :db.part/db}

{:db/id #db/id[:db.part/db]
 :db/ident :user/phone-number
 :db/valueType :db.type/ref
 :db/cardinality :db.cardinality/many
 :db.install/_attribute :db.part/db}
```

# Schema

```
{:db/id #db/id[:db.part/db]
 :db/ident :user/first-name
 :db/valueType :db.type/string
 :db/cardinality :db.cardinality/one
 :db.install/_attribute :db.part/db}

{:db/id #db/id[:db.part/db]
 :db/ident :user/age
 :db/valueType :db.type/long
 :db/cardinality :db.cardinality/one
 :db.install/_attribute :db.part/db}

{:db/id #db/id[:db.part/db]
 :db/ident :user/phone-number
 :db/valueType :db.type/ref
 :db/cardinality :db.cardinality/many
 :db.install/_attribute :db.part/db}
```

# Schema

```
{:db/id #db/id[:db.part/db]
 :db/ident :user/phone-number
 :db/valueType :db.type/ref
 :db/cardinality :db.cardinality/many
 :db.install/_attribute :db.part/db}
```

# Schema

```
{:db/id #db/id[:db.part/db]
 :db/ident :user/phone-number
 :db/valueType :db.type/ref
 :db/cardinality :db.cardinality/many
 :db/isComponent true
 :db.install/_attribute :db.part/db}
```

# Hands On

```
git clone https://github.com/jocrau/day-of-datomic.git
cd day-of-datomic
lein trampoline repl
```

Now: Emacs! No, VIM! No, Emacs! No, IntelliJ! No, LightTable!

# Queries

```
(d/q '[:find [?e ...]   SELECT
       :in $ ?email
       :where [?e :user/email ?email]]
     db "editor@example.com")
```

```
:find ?e              set of vectors
:find [?e …]          set of scalar values
:find [?e ?email]     tuple
:find ?e .            single scalar value
```

# Queries

```
(d/q '[:find [?e ...]
       :in $ ?email          input values
       :where [?e :user/email ?email]]
     db "editor@example.com")
```

# Queries

```
(d/q '[:find [?e ...]
       :in $ ?email     the database(s)
       :where [?e :user/email ?email]]
     db "editor@example.com")
```

```
$
$database
$db1 $db2
```

# Queries

```
(d/q '[:find [?e ...]
       :in $ ?email
       :where [?e :user/email ?email]]
     db "editor@example.com")
```

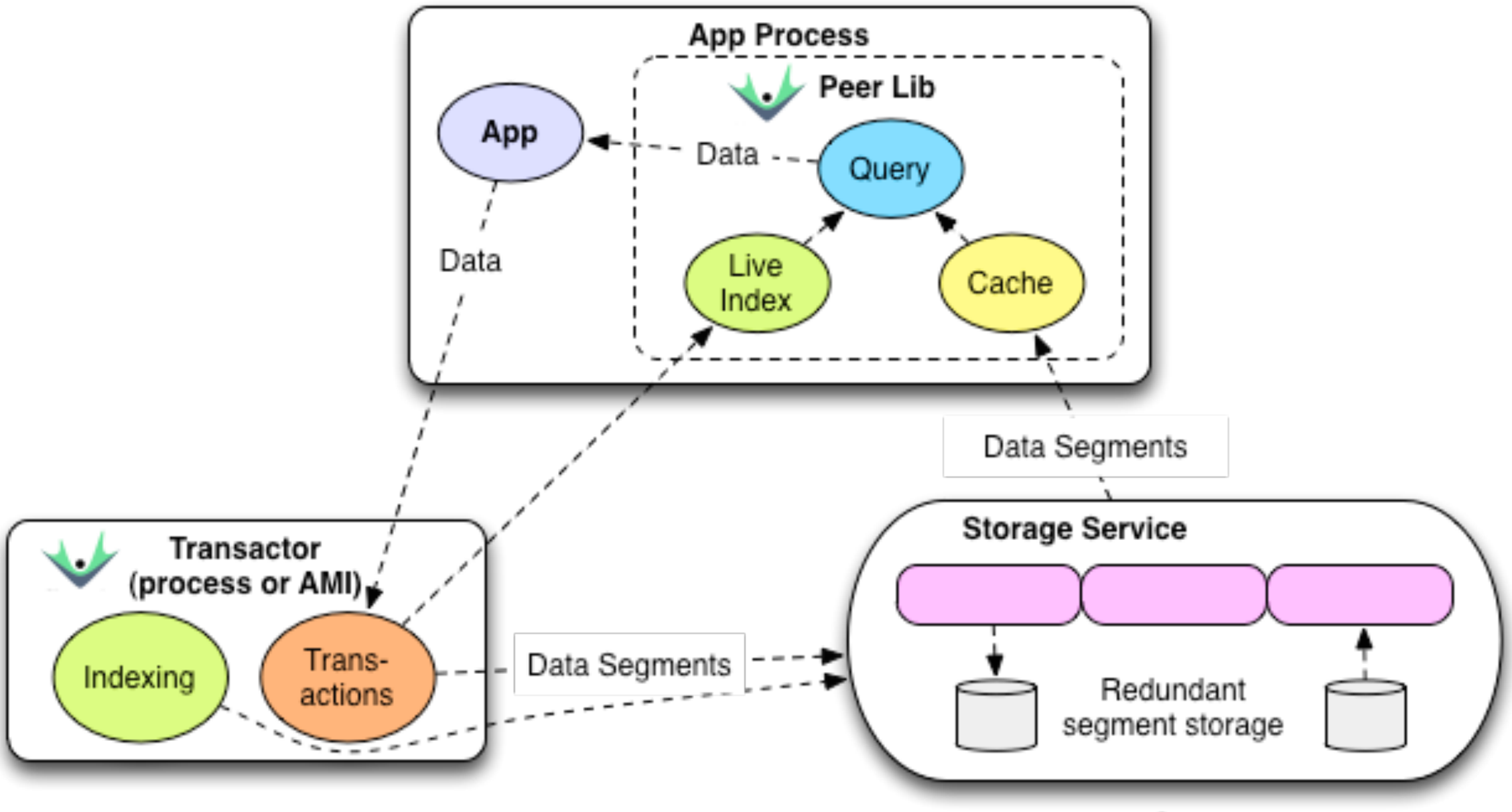
bound variable(s)

# Queries

```
(d/q '[:find [?e ...]
       :in $ ?email     data pattern(s)
       :where [?e :user/email ?email]]
     db "editor@example.com")
```

```
[?e :user/email ?email]
[?e :user/email]
[?e :user/email "editor@example.com"]
[?e ?attr "John"]
[(count ?email) ?email-length]
```

# Operational Structure



# Storage Services

- JDBC (MariaDB, PostgresSQL, Oracle, etc.)
- Amazon DynamoDB
- Basho Riak
- Apache Cassandra
- RedHat Infinispan (Immutant->JBoss)
- local disk (dev mode)

# SQL Backend Setup

- create a new SQL database
- create SQL table datomic_kvs (SQL script included)
- create SQL user and grant sufficient rights
- add JDBC driver dependencies (datomic/lib/ folder)
- configure transactor (datomic/config/transactor.properties)
  - JDBC connection string
  - license key

# Use Datomic

- for semantically rich, structured data
- if your application is read intensive
- if you need to add meta-data (access control)
- if you want to preserve history (audit trail)
- need flexibility (schema, operations)
- Online Transaction Processing (OLTP), Web Content Management Systems (WCMS), etc.

# Don’t Use Datomic

- for unstructured or binary data  (images, videos, texts)
- if your application is write intensive
- Stream Processing, (really) Big Data, Online Analytical Processing (OLAP) — well let’s talk about that ;-)

# What’s missing

- data type for clojure data structures (or at least for queries)
  - traversable or non-traversable
  - enables higher-order queries
- adding attributes on the fly
- open vs. closed world assumption
- support for reasoning (schema->ontology->reasoner)

# Thanks!

```
{:db/id                  #db/id [:db.part/user -300015]
 :resource/id            #uuid "16447494-fa8c-4f6f-90d4-6a68eff9d7c8"
 :xnl/person-name        [{:xnl/person-name-type :xnl.person-name-type/LegalName
                           :xnl/name-element [{:content/value "Jochen"
                                               :xnl/element-type :xnl.element-type/FirstName}
                                              {:content/value "Rau"
                                               :xnl/element-type :xnl.element-type/LastName}]}]
 :coho/membership        [{:coho/party #db/id [:db.part/user -200023]
                           :coho/valid-from #inst "2012-11-01T00:00:00+00:00"}]
 :xpil/birth-info        {:xpil/birth-date-time     #inst "1971-03-15T00:00:00+00:00"
                          :coho/date-time-precision :coho.date-time-precision/DayMonthYear}
 :xpil/electronic-address [{:xpil/identifier "jochen@rau-family.us"
                            :xpil/service :xpil.service/Email
                            :xpil/usage :xpil.usage/Private}
                           {:xpil/identifier "jrau@infinitecloud.com"
                            :xpil/service :xpil.service/Email
                            :xpil/usage :xpil.usage/Business}
                           {:xpil/identifier "jocrau"
                            :xpil/service :xpil.service/Skype
                            :xpil/usage :xpil.usage/Private}]
 :xpil/telephone-number   [{:xpil/area-code "413"
                            :xpil/exchange-code "695"
                            :xpil/local-number "2315"
                            :xpil/usage [:xpil.usage/Private :xpil.usage/Primary]
                            :xpil/media-type :xpil.media-type/Mobile}]}
```