

**FQL**

query language for Foundation DB

**What is Foundation DB?**

**Open Source**

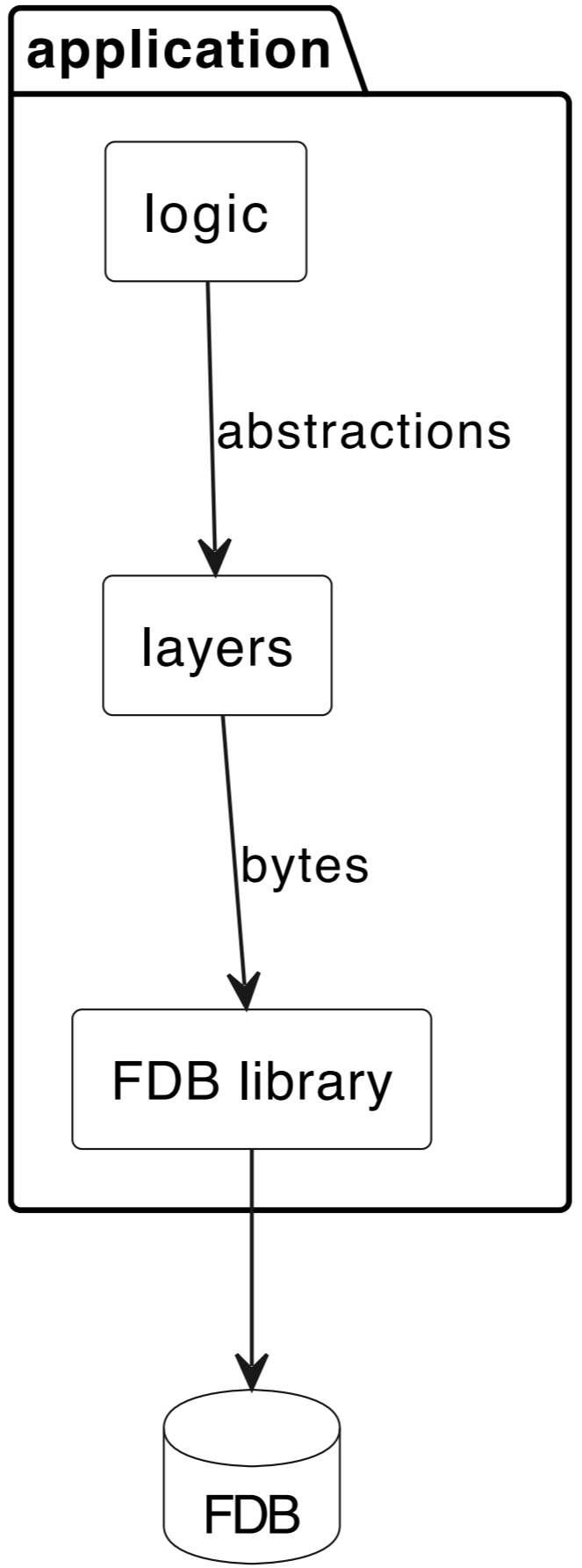
**Distributed**

**Automatic**

**ACID**

```
@fdb.transactional
def write_stuff(tr):
    for k, v in data.items():
        tr[k] = v # write the KV
```

```
07d8bd76955f1dc6 -> 81dd3582cabd853f7fd4d2
59b7746eb4bef303 -> 9680dcc57eee76bf75cfda
9bc1757c95034e4f -> f69f6a6f971e245a36870f
b8a0f6e2896502cb -> 82d2a760b4dfbd0e4659ff
c7ae3b03cd4a549b -> 2c157fa9623ae5fcfa731d
cba5b5c8012c1db4 -> d4d0cb63d5d9fd94a96214
f918b1210d84f1cc -> 717530d6f1a8f6be061665
```



```
path = ["regions", "na"]

@fdb.transactional
def write_stuff(tr):
    d = dir.open(tr, path)
    for k, v in data.items():
        tr[d.key().append(k)] = v
```

```
[regions, na] 6eb4bef303 -> ...
[regions, na] 76955f1dc6 -> ...
[regions, na] 7c95034e4f -> ...
[regions, na] e2896502cb -> ...
[regions, af] 03cd4a549b -> ...
[regions, af] 210d84f1cc -> ...
[regions, af] c8012c1db4 -> ...
```

```
keys = [  
    (10, "hi", 99.8),  
    (3, "y"),  
    (None, ("sub")),  
    (3, "a", true)  
]  
  
@fdb.transactional  
def write_stuff(tr):  
    for key in keys:  
        tr[tuple.pack(key)] = None
```

```
(None, ("sub")) -> None  
(3, "a", true) -> None  
(3, "y") -> None  
(10, "hi", 99.8) -> None
```

```
# parts of a key-value...
path = ["regions", "na"]
tup = ("hello", 42)
val = pickle.dumps(obj)

@fdb.transactional
def write_stuff(tr):
    d = dir.open(tr, path)
    tr[d.pack(tup)] = val
```

**Why a query language?**



# Basic Queries

```
/my/dir(99.8, "hello")=42
```

```
/entries(  
  1c389909-0843-4e2a-a351-8a5d90d8b3d2,  
  true,  
)="somewhere"
```

```
/pairs/bonded(2353218923) =  
("james", "ricky")
```

```
/my/dir (<num|bool>, "hello") =<>
```

```
% these key-values match the schema...
```

```
/my/dir (22.354, "hello") = nil
```

```
/my/dir (true, "hello") = 0xde37
```

```
% these key-values don't match the  
schema...
```

```
/other/dir (22.3102, "hello") = 0x337a
```

```
/my/dir (-7.2, "hello", "world") = nil
```

```
/my/numbers (32, ...) =<nil|bytes>
```

```
% these key-values match the schema...
```

```
/my/numbers (32) =nil
```

```
/my/numbers (32, 102, 103, 104) =0xffab
```

```
/my/numbers (32, 3203) =nil
```

```
% write  
/entries/quotes(753985)="seize the day"
```

```
% clear  
/entries/quotes(753985)=clear
```

```
% single read  
/staff/admin(1233872)=<num>
```

```
% range read  
/users(...)=<>
```

```
% directories  
/home/documents/<>/archive
```

nil

true

257

33.9

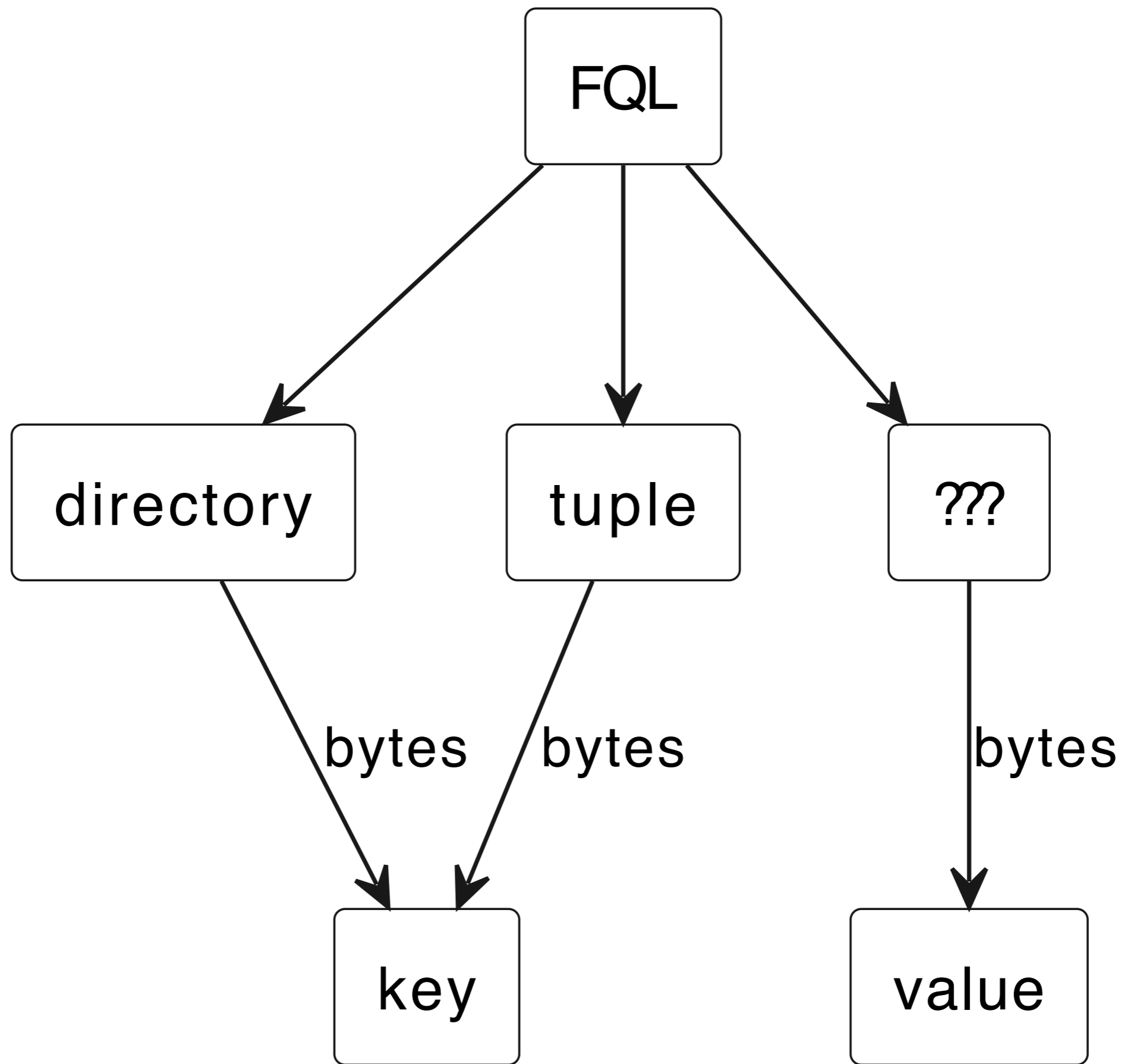
"string"

#123072341428501298312781234089123

0x96f08a3cffb4767284b3d5f9760ebd98

67b91324-d8cb-4f6c-83ef-2e491ba50527

("hello", 27.4, nil)



```
/my/dir("my", "key") =<int|uuid|nil>
```



# **Advanced Queries**

```
/people (  
    <uint>, % ID  
    <str>, % first name  
    <str>, % last name  
)=nil
```

```
/people (1234123, "Larry", "Johnson")=nil  
/people (3253234, "John", "Rogers")=nil  
/people (5423432, "Wilson", "Johnson")=nil
```

```
/people/last_name(  
    <str>,    % last name  
    <uint>,  % ID  
)=nil
```

```
/people/last_name("Johnson",1234123)=nil  
/people/last_name("Johnson",5423432)=nil  
/people/last_name("Rogers",3253234)=nil
```

```
/people/last_name("Johnson",  
<id:uint>)=nil  
/people(:id,...)=nil
```

```
/people(1234123,"Larry","Johnson")=nil  
/people(5423432,"Wilson","Johnson")=nil
```

```
/blob(  
    "large_file.bin", % name of the blob  
    <int>,           % byte offset within  
the blob  
)=<bytes>         % chunk of the blob
```

```
/blob("large_file.bin",...)=<>
```

```
/blob("large_file.bin",0)=10000_bytes  
/blob("large_file.bin",10000)=10000_bytes  
/blob("large_file.bin",20000)=5320_bytes
```

```
/blob("large_file.bin",...)=<append>
```

```
/blob("large_file.bin",...) = 2532_bytes
```

> .

<https://github.com/janderland/fql>

<https://jander.land>