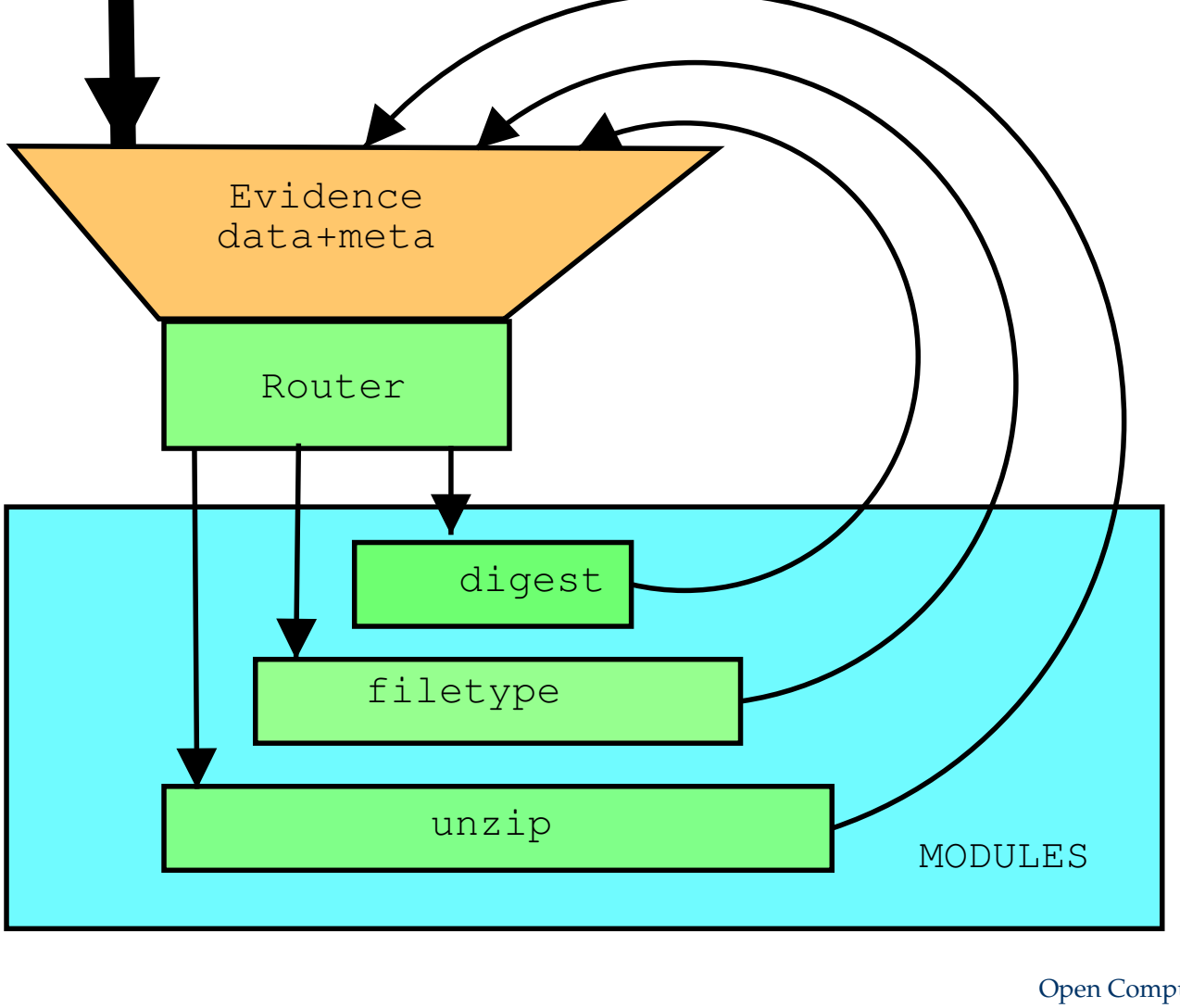
Open Computer Forensics Architecture API

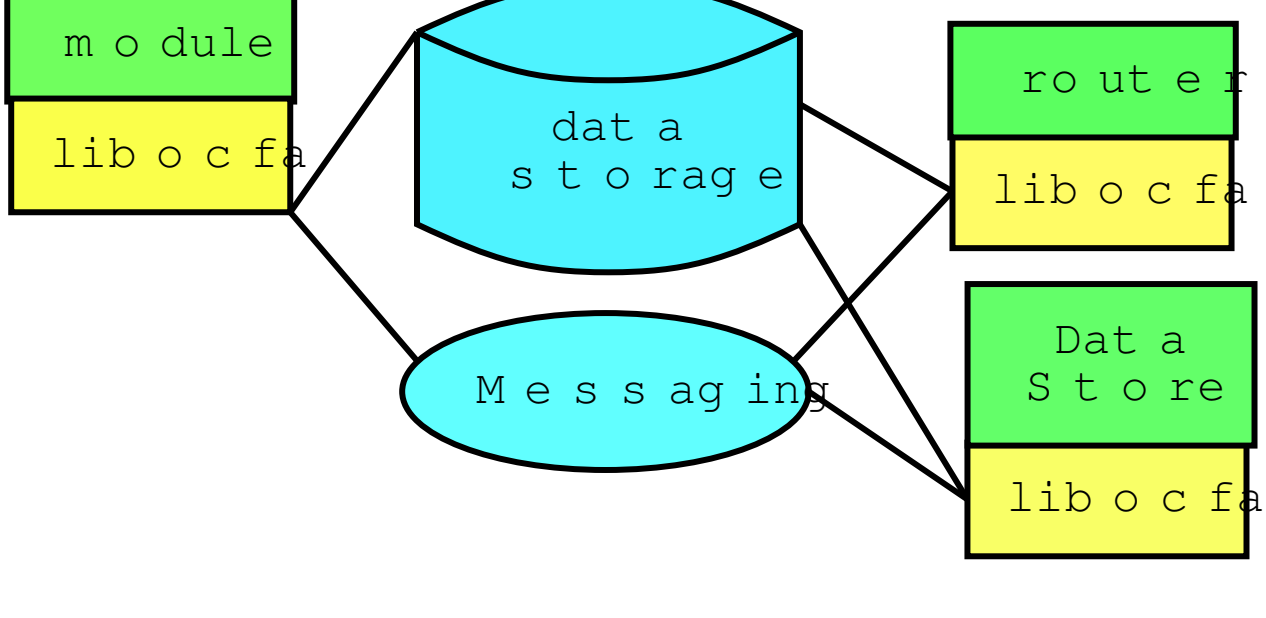
Rob J Meijer

Korps Landelijke Politiediensten, Driebergen, NL

Digital Washing Machine



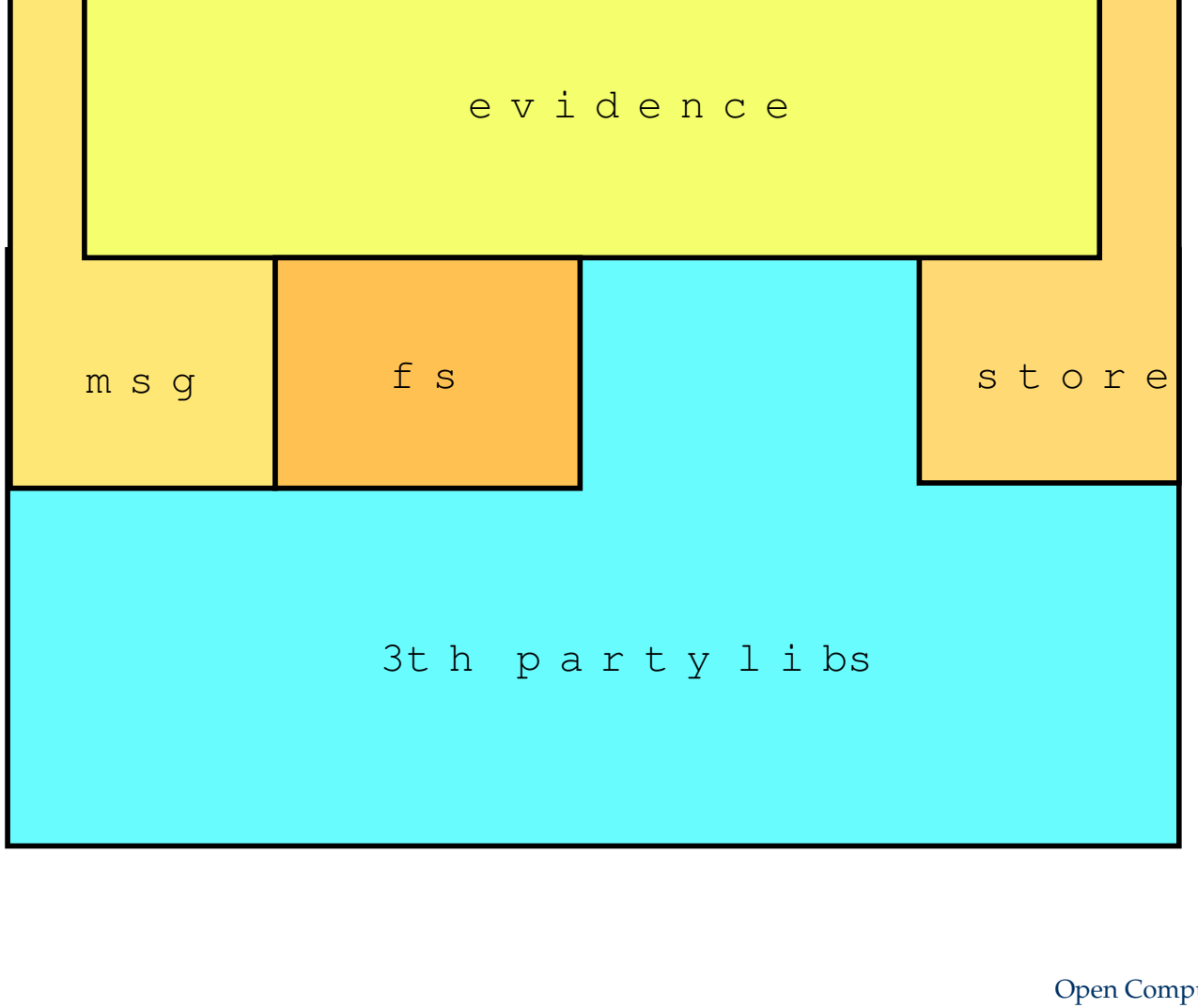
Open Computer Forensics Architecture



Dual module/architecture interface

- OCFA Evidence API
- Evidence XML, OCFA Message/Store API

LibOcfa



Dependencies on 3th party libs

Evidence:

- libcrypto (digests)
- iconv (charsets)
- xerces (XML)

Store/Msg:

- libpq++ (postgress database)

fs/store:

- posix (libgen,unistd,stat)

XML: Evidence,Jobs & Metadata

XML : evidence

Attributes:

- ID Fields (case,source,item,id)
- Digests (SHA1,MD5)
- status (ACTIVE / SUSPENDED / COMMITTED)

Members:

- location
- job

XML : job

Attributes

- timestamps (stime,etime)
- status (NEW/PROCESSED/DONE)

Members

- ModuleInstance
- Meta
- LogLine

XML: meta & scalar

Meta Attributes

- Name
- Type (Scalar,Array,Table)

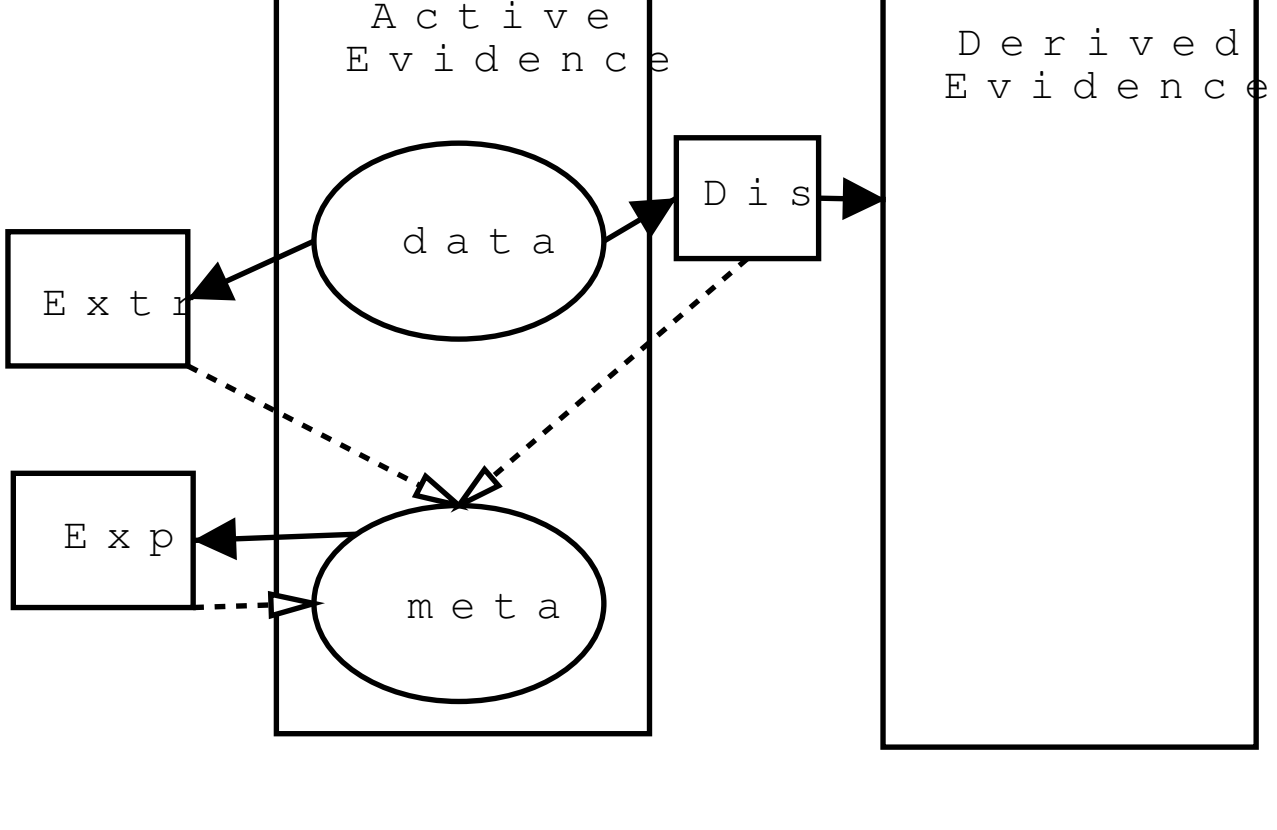
Scalar Attributes

- Type (Int,Float,string,datetime)

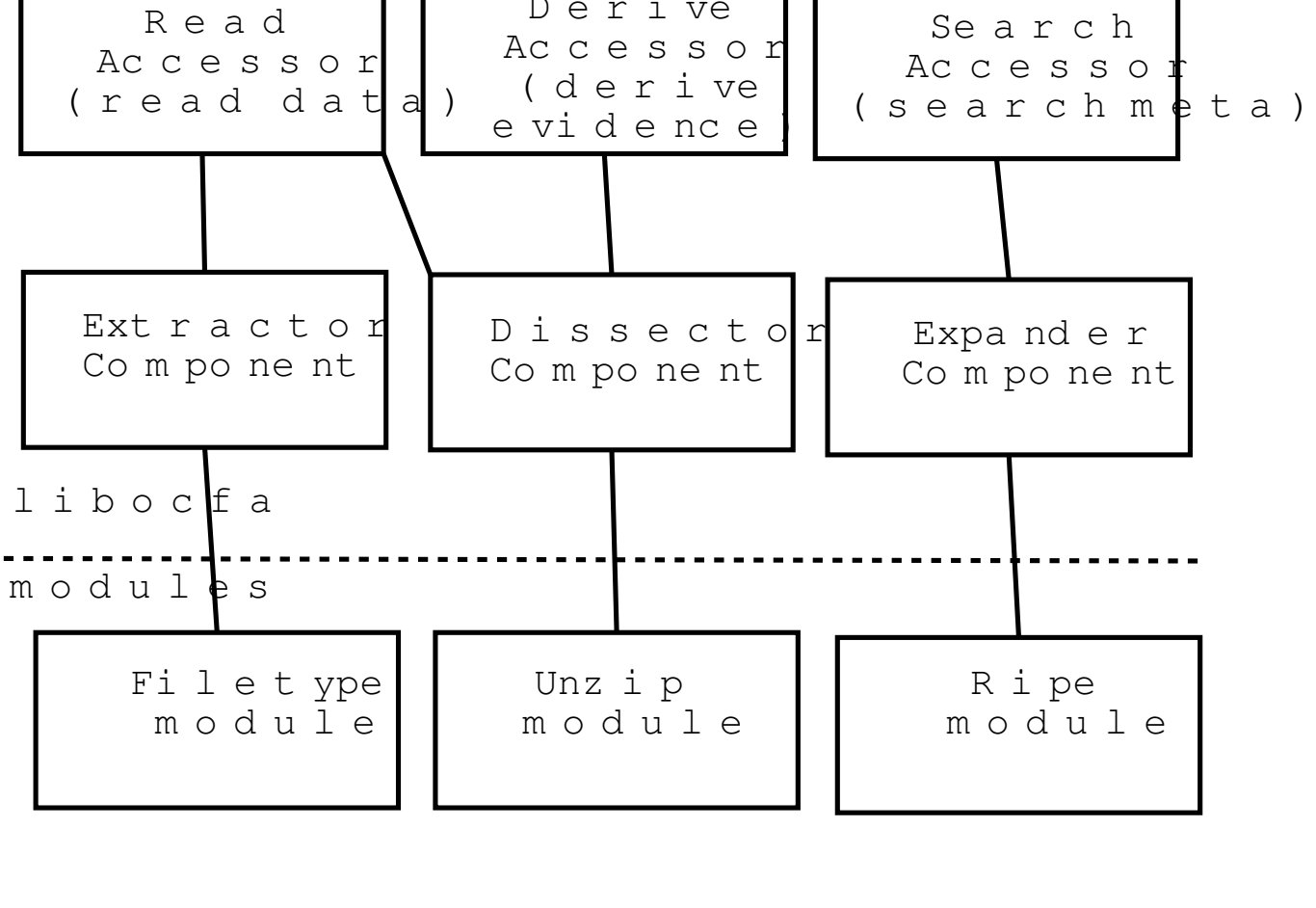
Scalar Content

- Int : long long
- Float : long double
- String : UTF8
- datetime: unix time + sourceref

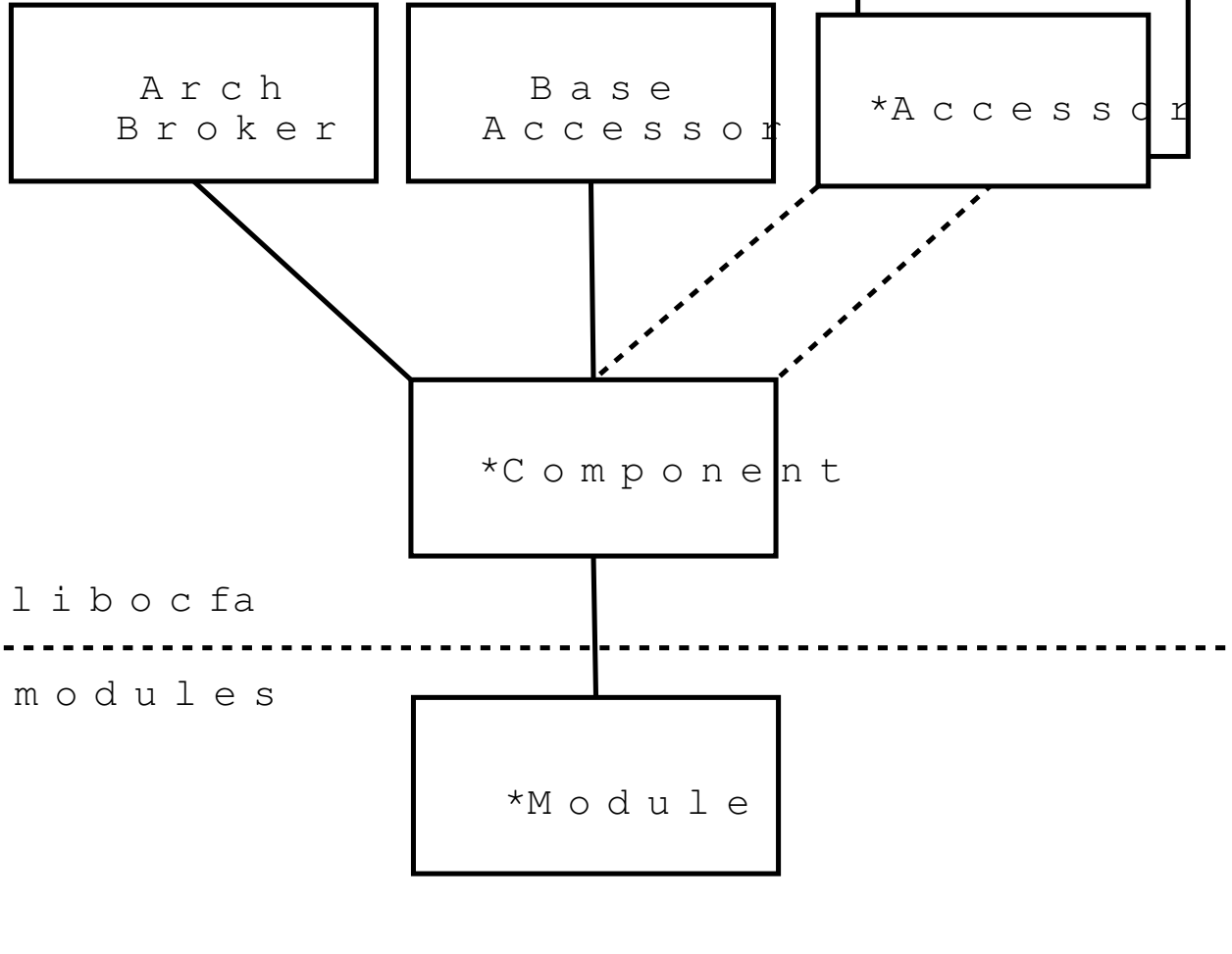
Module Types



Types & Methods



Inheritance



*Accessor

- Additional 'protected' methods
- Access to method groups of 'ActiveEvidence
- Access to method groups of 'ActiveJob'

ArchBroker

- virtual int processEvidence()
- virtual void processMessage()
- void run()

Module main

```
int main(int argc,char *argv[]){
  xExtractor *x=new xExtractor();
  x->run();
}
```

BaseAccessor

- void setMeta(name,value)
- void pushBackMeta(name,value)
- Scalar getEvidenceLocation()
- string getDigestSHA1()
- string getDigestMD5()

BaseAccessor example

```
int DigestModule::processEvidence()
  string sha1=getDigestSHA1();
  string dsrc=somedblookupmethod(
    sha1);
  setMeta("digestsource",
    Scalar(dsrc));
  return 1;
}
```

ReadAccessor

- EvidenceStoreEntity *getEvidenceStoreHandle()

ReadAccessor example

```
EvidenceStoreEntity *estore=
  getEvidenceStoreHandle();
snprintf(cline,MAXLINE,
 "gpg -v --list-packets --batch %s",
 estore->getAsFilePath().c_str());
command=popen(cline,"r");
```

DeriveAccessor

- string getWorkDir()
- DerivedEvidence *derive(content,name,..)

DeriveAccessor example

```
DerivedEvidence *derived=
  derive("/output.txt",
    Scalar("[output]"));
derived->setMeta("mimetype",
  Scalar("text/x-ocfa-strings"));
derived->submit();
delete derived;
```

'Open' Computer Forensics Architecture ??

- Open API !
- Open XML Schema !
- Reference implementation.
  - Open to LEAs !
  - Open to Vendors ?
  - Open to the general public ?