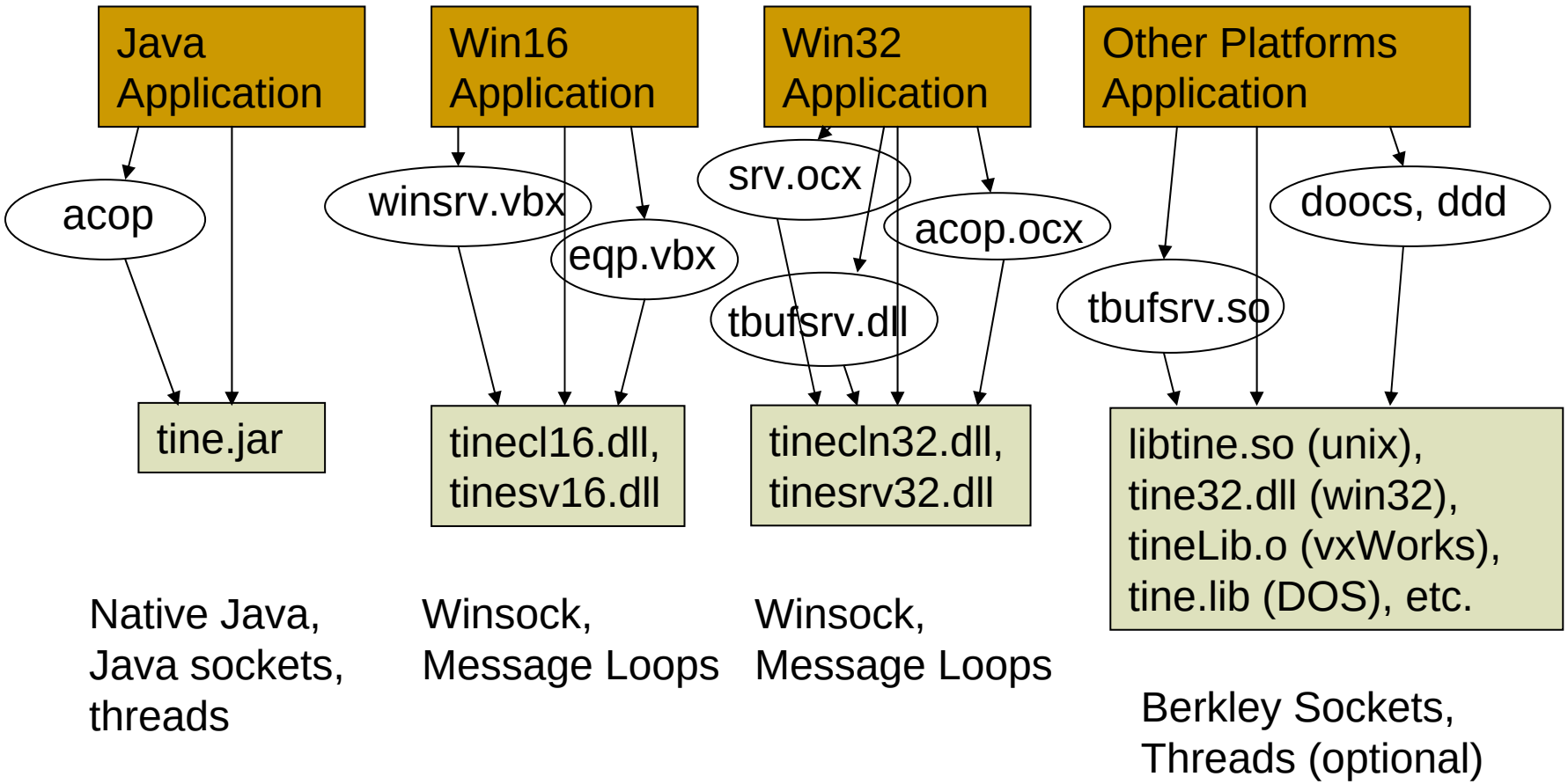
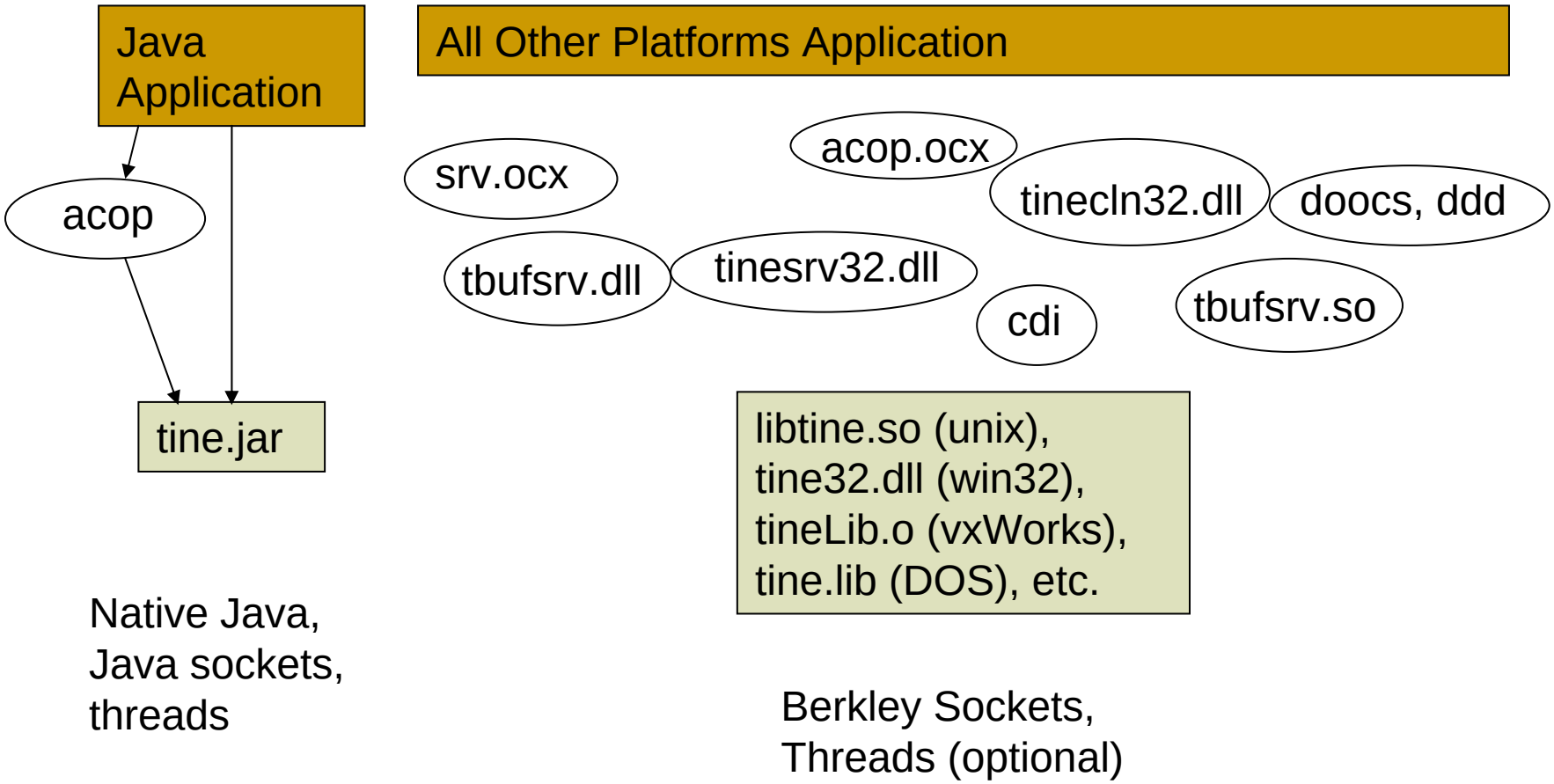
TINE Release 4.0 Status

What's been going on with all of those 'new' dlls ?

Overview Libraries (before)



Overview Libraries (after)



TINE Release 4.0.0 Highlights

- Allowed Name lengths greatly increased!
  - Registered Device Names, Properties -> 64 chars
    - Device Name String up to 1025 chars
    - e.g. “M1Adc.rstTrg,M3Adc.rstTrg,M5Adc.rstTrg,…”
  - Structure,Bitfield Tags up to 16 chars
  - Registered Context, Device Server names -> 32 chars
- Case Insensitivity
  - e.g. No difference between “TEST” and “Test”
  - e.g. No difference between “NR 64 MO” and “nr 64 mo”
  - e.g. No difference between “RESET” and “reset”

TINE Release 4.0.0 Highlights

- New Data Formats
  - CF_XML
    - Sent as a text string
  - CF_VIDEO
    - Video header + frame
  - CF_BITFIELD8, CF_BITFIELD16, CF_BITFIELD32, CF_BITFIELD64
    - Data type: DBITFIELD
      - bitfield segments from 1 bit to full range have names
    - Bitfield Registry
      - e.g. addFieldToBitField("thisfec","StsBits",0xf0,"field3");
      - Property "Status" registered with format CF_BITFIELD16
      - "Status.field3" gives 2nd Nibble of the Status Word!
- Tagged structures:
  - Can now contain (arrays of) any allowed data type, including other (registered) tagged structures (e.g.)
    - Struct

```
{
  float value;
  int    status;
  RANGE range;
  FLTINT rawdata;
}
```

TINE Release 4.0.0 Highlights

- Server Configuration
  - API Configuration as before
    - RegisterFecInformation(), RegisterProperty(), etc.
  - .csv Configuration as before
    - FEC_HOME -> fecid.csv
      - Subdirectories for Equipment Modules
        - exports.csv, history.csv, alarms.csv, devices.csv
      - 'HISTORY_HOME' column in fecid.csv supersedes environment variable
  - **.xml Configuration !!**
    - Single xml file : fec.xml

fec.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
- <COMPUTER>
  - <FEC>
      <NAME>MSTXPDUVAL03.23</NAME>
      <PORT_OFFSET>23</PORT_OFFSET>
    - <EQM>
        <NAME>BPMEQM</NAME>
        <SERVER>DoBeam</SERVER>
        <CONTEXT>DORIS</CONTEXT>
        <SUBSYSTEM>DIAG</SUBSYSTEM>
        <DEVICE_SPACE>42</DEVICE_SPACE>
      - <DEVICE>
          <NAME>NR 3 MO</NAME>
        </DEVICE>
      - <DEVICE>
          <NAME>NR 6 MO</NAME>
        </DEVICE>
      - <DEVICE>
          <NAME>NR 7 MO</NAME>
        </DEVICE>
      - <PROPERTY>
          <ID>1</ID>
          <NAME>OrbitX</NAME>
          <DESCRIPTION>[-10:10 mm]Horizontal Orbit</DESCRIPTION>
          <SIZE_OUT>42</SIZE_OUT>
          <DTYPE_OUT>float.CHANNEL</DTYPE_OUT>
          <ACCESS>READ</ACCESS>
          <REDIRECTION />
        </PROPERTY>
      - <PROPERTY>
          <ID>2</ID>
          <NAME>OrbitY</NAME>
          <DESCRIPTION>[-10:10 mm]Vertical Orbit</DESCRIPTION>
          <SIZE_OUT>42</SIZE_OUT>
          <DTYPE_OUT>float.CHANNEL</DTYPE_OUT>
          <ACCESS>READ</ACCESS>
          <REDIRECTION />
        </PROPERTY>
      </EQM>
```

TINE Release 4.0.0 Highlights

- Expanded Data Object (DTYPE)
  - dArrayLength (as before)
  - dFormat (as before)
  - dTimeStamp (as before)
  - dTag (as before, but now longer)
  - **dStamp** (a user supplied integer tag)
  - **sysStamp** (a systematic integer tag : e.g. cycle number, run number)
  - **xferReason** :
    - CX_NULL, CX_RESPONSE, CX_STALE, CX_HEARTBEAT, CX_EVENT, CX_TIMER, etc.

TINE Release 4.0.0 Highlights

- Expanded Alarm Message Structure
  - timestamp (secs + **usecs**)
  - **starttime** (**secs + usecs**)
  - code (as before)
  - status (as before)
  - data (**64 bytes**! – was 6)

TINE Release 4.0.0 Highlights

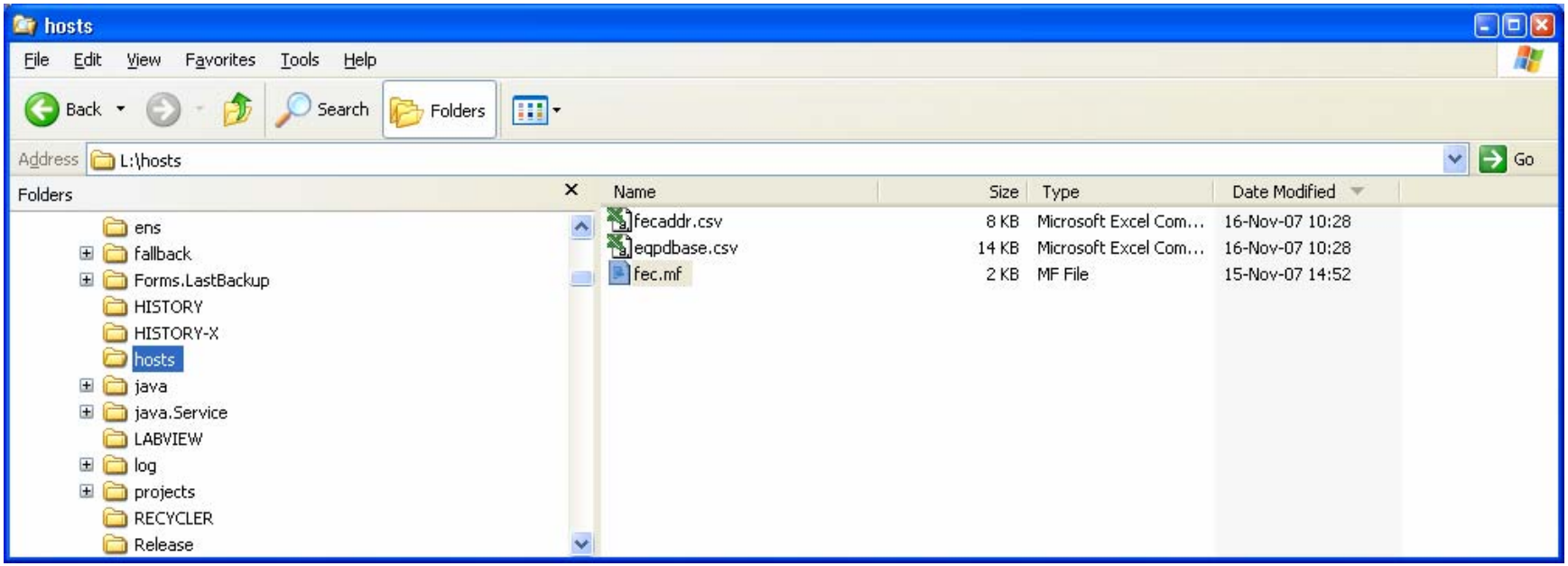
- Dynamic Client-side Name Caching
  - Name resolution:
    - First ask the configured ENSes
    - Then consult the dynamic Name cache
    - Then consult the static Name cache (if present)
  - Once a Client has acquired an Address the local dynamic cache is updated!
  - Upon ENS failure, the last known address is probably as good as anything else!

Dynamic cache

On Windows: %SystemDrive%:\tine\hosts

On Unix: /tmp/tine/hosts

%SystemDrive:\tine usually mapped as L:



Work Station manifest

	EXPORT	CONTEXT	EQM	FEC	PORT	VERSION	FEC_HOME	LAST_STARTED
2	NETWORK	SERVICE	_SRV__	ENS	0	4.00.0000		Thu Nov 15 14:52:08 2007
3	ENS	SERVICE	ENSEQM	ENS	0	4.00.0000		Thu Nov 15 14:52:08 2007
4	VbSineServer	TEST	SINEQM	VBSINESRV.18	18	4.00.0000	c:\tine\database\server\	Wed Nov 14 18:28:31 2007
5	WinSineGen	TEST	SINEQM	MSTXPDUVAL03.18	18	4.00.0000	c:\tine\database\server\	Sat Nov 03 12:23:12 2007
6	MSTXPDUVAL03_WD	SERVICE	WD_L03	MSTXPDUVAL03.20	20	4.00.0000	c:\tine\database\server\	Fri Nov 02 09:42:39 2007
7	MSTNTSINEX		WINEQM	MSTXPDUVAL03.16	18	4.00.0000	c:\tine\database\server\	Thu Nov 08 09:24:00 2007
8	MSTXPDUVAL03_WD		WD_L03	MSTXPDUVAL03.18	18	4.00.0000	c:\tine\database\server\	Fri Nov 02 09:22:32 2007
9	MSTXPDUVAL03.CDI	HARDWARE	CDIEQM	MSTXPDUVAL03	30	4.00.0000	c:\tine\database\server\	Fri Nov 02 13:13:28 2007
10	BUFSINE	TEST	WINEQM	MSTXPDUVAL03.16	18	4.00.0000	c:\tine\database\server\	Thu Nov 08 16:40:04 2007
11	GTTEST	TEST	GRAEQM	*unknown*	11	4.00.0000	c:\tine\database\server\	Thu Nov 08 14:45:18 2007
12	GTTESTF	TEST	GRFEQM	GTTEST	11	4.00.0000	c:\tine\database\server\	Thu Nov 08 14:44:52 2007
13	MSTNTSINEX	TEST	WINEQM	MSTXPDUVAL03.16	18	4.00.0000	c:\tine\database\server\	Mon Nov 12 18:57:07 2007
14	TVWORKSHOP		GRAEQM	*unknown*	0	4.00.0000	c:\tine\database\server\	Thu Nov 08 17:26:41 2007
15	TVWORKSHOPF		GRFEQM	*unknown*	0	4.00.0000	c:\tine\database\server\	Thu Nov 08 17:26:46 2007
16	WinSineServer		SINEQM	MSTXPDUVAL03.18	18	4.00.0000	c:\tine\database\server\	Mon Nov 12 18:16:26 2007

TINE Release 4.0.0 Highlights

- Revised Multicast Address Scheme (Kars Ohrenberg)
  - Globals multicast (Producers)
  - Publisher multicast
  - Services multicast
- Classic (old) way:
  - Each has a single multicast group
- Standard (new) way:
  - Each server on the control net has its own multicast group!
  - No more ‘N-Producer’ problem!
  - Services multicast still uses a single systematically known multicast group.

TINE Release 4.0.0 Highlights

- Local history system to use “worst-case” non-fragmented files
- Time Synchronization to 100 msec.
  - (Requires client-side daemon ?)
- Forced transfer efficiency of multi-channel arrays, bitfields, user-defined structures.
- Adjustable Local History, Alarm settings from remote location.

API Breaks (c):

- int GetCallerInfo(NAME16 *un,BYTE *ipx,UINT32 *ip,short *prot,int *num);
  ->
  int GetCallerInfo(**char *eqm**,NAME16 *un,BYTE *ipx,UINT32 *ip,short *prot,int *num);
- char *GetCaller(void);
  ->
  char *GetCaller(**char *eqm**);
- void SetRPCCompletion(char *errstr)
  ->
  void SetEqmCompletion(**char *eqm**, char *errstr)
- int AssignPropertyList(char *eqm,char *devname,char *listname,int listsize,**NAME64** *list)
- int GetRegisteredPropertyList(char *eqm,**NAME64** *prpNames,int *nprps);
- Some legacy calls no longer supported (e.g “RPC()”)

Release 4.0 at DESY

- Primary and Secondary ENS are running release 4.0 (since Tuesday)
- Office Windows PCs are running Release 4.0 Dlls (almost 2 weeks – with sometimes hourly updates of DLLs)
- Several Servers (thanks Mark) are running and testing Release 4.0 server compatibility
- Current strategy is to make sure what used to work still works and test new features afterwards.

Ways of finding the ENS

- There's a cshost.csv file on the local file system
  - Can also contain a single column TINE_HOST with the host name of the computer running the ENS (idea from Gunter Trowitch).
  - An environment variable points to this location of this file (TINE_HOME) or (Java) a property tine.home points to this location.
  - The file is in the working directory
- There's an environment variable TINE_ENS="131.169.120.41,131.169.120.146"
- The application sets the address with an API call
- The application sends a multicast to find the ENS
- The application as a last resort (Release 4.0) looks for a host called tineens on the current domain.

Today's Zustand

- Cdi library is linked to the release 4.0 TINE library but does NOT make use of long names at the moment!
- Windows Instant Client does not yet make query calls to get 'long' names
- BUT: Acop.ocx does!
- The most recent 'round of bugs/feature wishes' will be integrated into tine.jar before porting tine.jar to Release 4.0

TODO (Kernel):

- Some stock Properties (e.g. a history call) on separate thread) -> almost works
- Allow tagged structure fields to be called as a property (with forced efficiency)
  - e.g. “SineCurve.Amplitude” (whereby the entire struct Property “SineCurve” is retrieved (under the hood).
- Force efficiency in multi-channel array calls
  - e.g. calling channel property “Pressure” for a single “Pump10” acquires the entire channel array.

TODO (tools):

- Local History File setup tool
- Time Sync daemon (if necessary)
- Fec configuration tool (a la FEC setup wizard)
  - Use to shell C-server wizard

TODO (Services):

- CAS
  - make use of new alarm message structure
- Central Archive
  - use a minimal file fragmentation logic
  - allow archive of compound data types
- Event Archive (Post Mortem)
  - New (self-describing?) data header structures
- Central Logger
  - In a test phase (needs to be implemented)
  - API is already in the kernel