

Reporting Tools With Tcl

History

- Start was about 1995 in payed work
- Client/Server architecture
- Originally use of socket interface
- Client as WISE installation package

Original Architecture

- Private protocol on socket interface
- Client uses Tcl/Tk
- Client is stored in files in a COMMON and an application specific directory
- After installation automatic update of client when starting
- Differences are identified by timestamp on server and client
- Out-of-date files are stored on the client and evaled in parallel

Goal of the Tool

- Present financial reports as tables
- Entering verified data to be used as SAP frontend
- Drive selection and contents of report with comboboxes
- There are also forms for entering data
- Contents of report may depend on access rights and function of users

Step 2

- Use of apache server with rivet on server side
- Use of http protocol to client
- Client is acting as a „browser“
- Adaption for client is at one place (http instead of socket protocol)

Step 3

- Use of starkit for client instead of WISE installation
- Only one .exe file necessary on client
- Starkit contains Tcl/Tk runtime, Itcl, Bwidget, Iwidgets and minimal client
- Specific client is built on info sent from server

Step 4

- Use of DB (mysql/oracle) for client metainfo
- Client is a generic client with generic parts like spreadsheet table, comboboxes, entryfields etc.
- Specific layout of client is driven by attributes to the elements of a „report“
- A „report“ can also be a form for entering data
- A „report“ can have a forms part and a spreadsheet part
- Selection of reports with tabnotebook later with a tree

Step 5

- Reimplementing the presented tools as open source project
- Use of newer features of Tcl/Tk and itclng
- Use of sqlite3 for layout/attribute infos
- Rebuild existing tool for administration of layout/attribute infos (this is built also with the same technique)
- Restructure/simplify layout/attribute info
- Enable use of different UI techniques like tile

General Structure

- Client very similar to existing client
- Use of paned windows one for selection of report from a tree, one for the form/report part
- Form/report part is a frame consisting of upto 9 frames within it
- Every part within it can again consist of upto 9 frames

Layout Details (1)

- Use of grid manager
- Allows multi rows/columns frames
- Allows selection box area to be vertical or horizontal
- Allows form for entering data and spreadsheet part in one „report“

Layout Details (2)

opleft

topcenter

topright

centerleft

centercenter

centerright

opleft

topcenter

topright

centerleft

centercenter

centerright

bottomleft

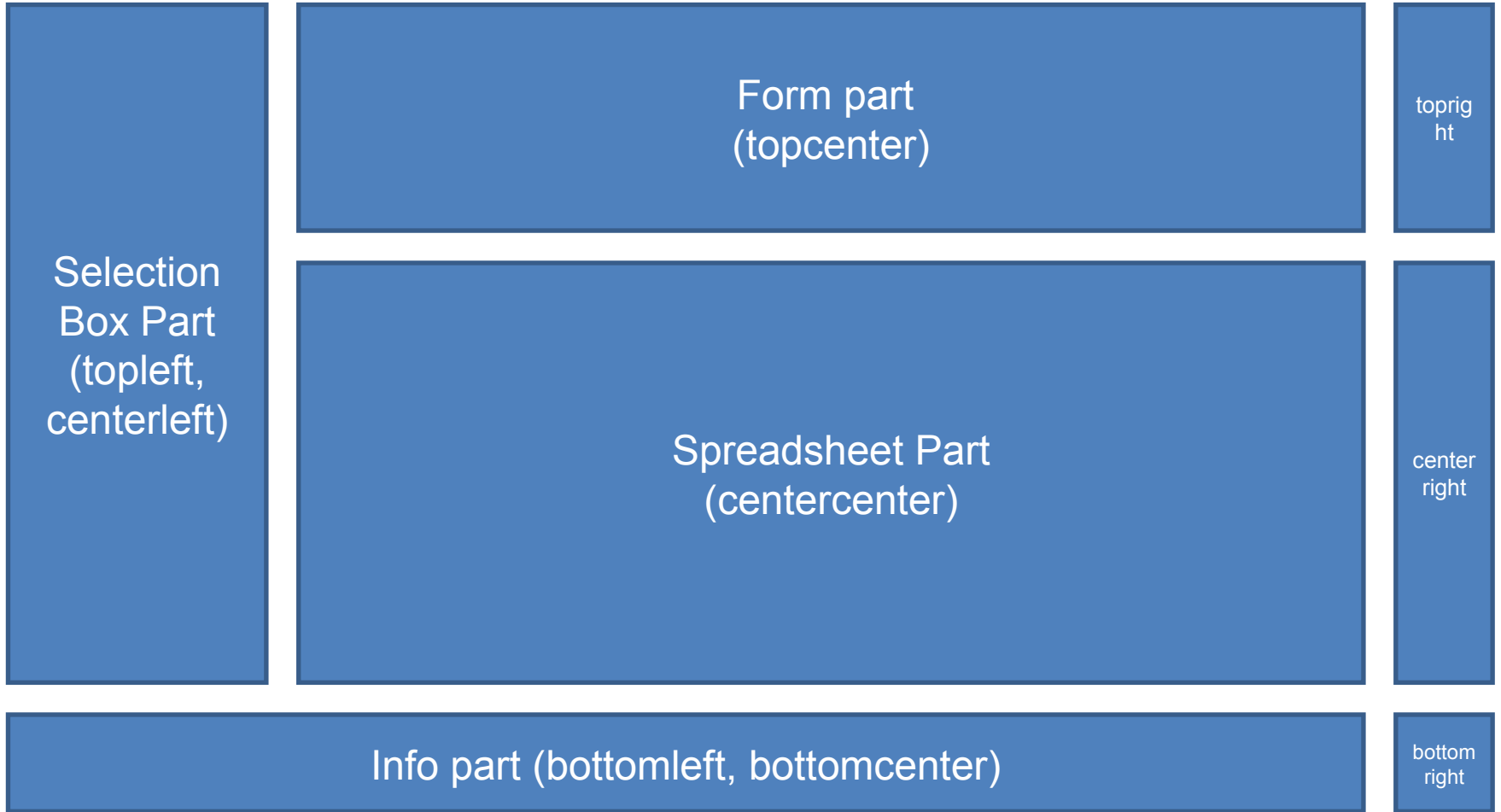
bottomcenter

bottomright

bottomcenter

bottomright

Layout Details (3)



Some UI Elements

Combobox

Entryfield

Calendar Widget

Radio Box

Button Box

Label

Table Report
(spreadsheet)

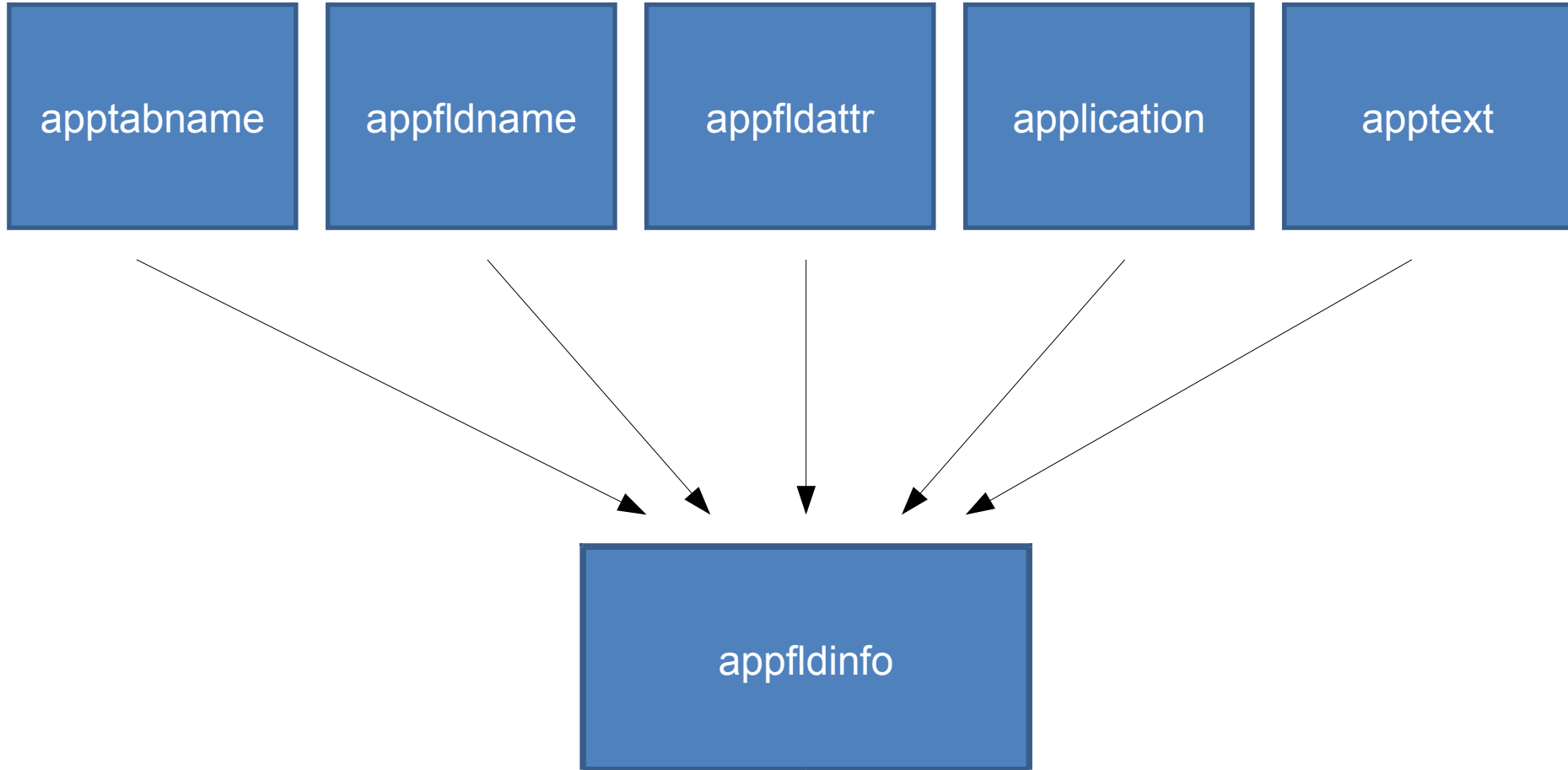
Input Report
(form for entering data)

Tree Report

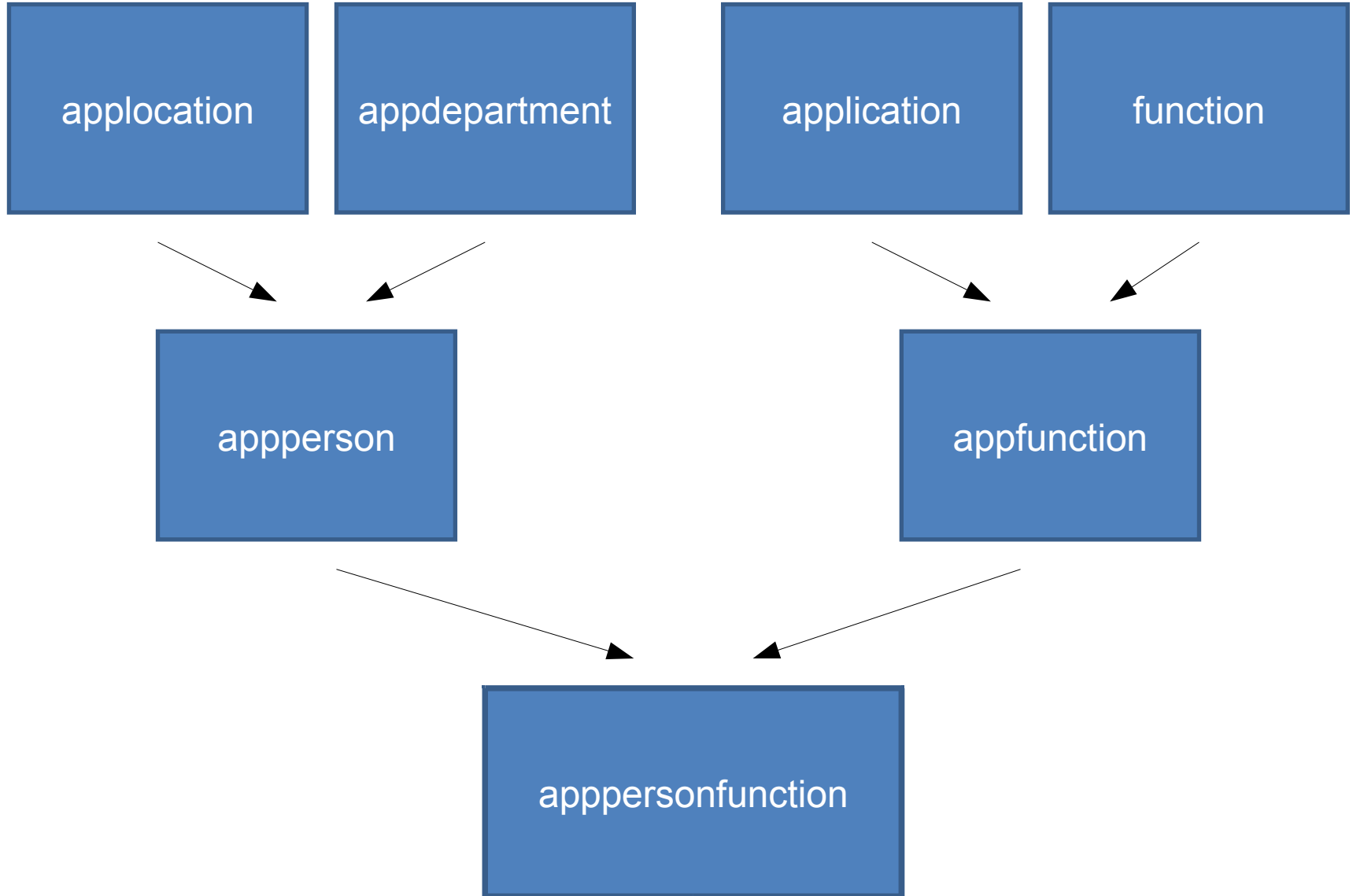
Attributes for UI Elements

- row
- column
- adjust
- text (language dependent label)
- label position
- sticky
- width
- height
- fixed
- :

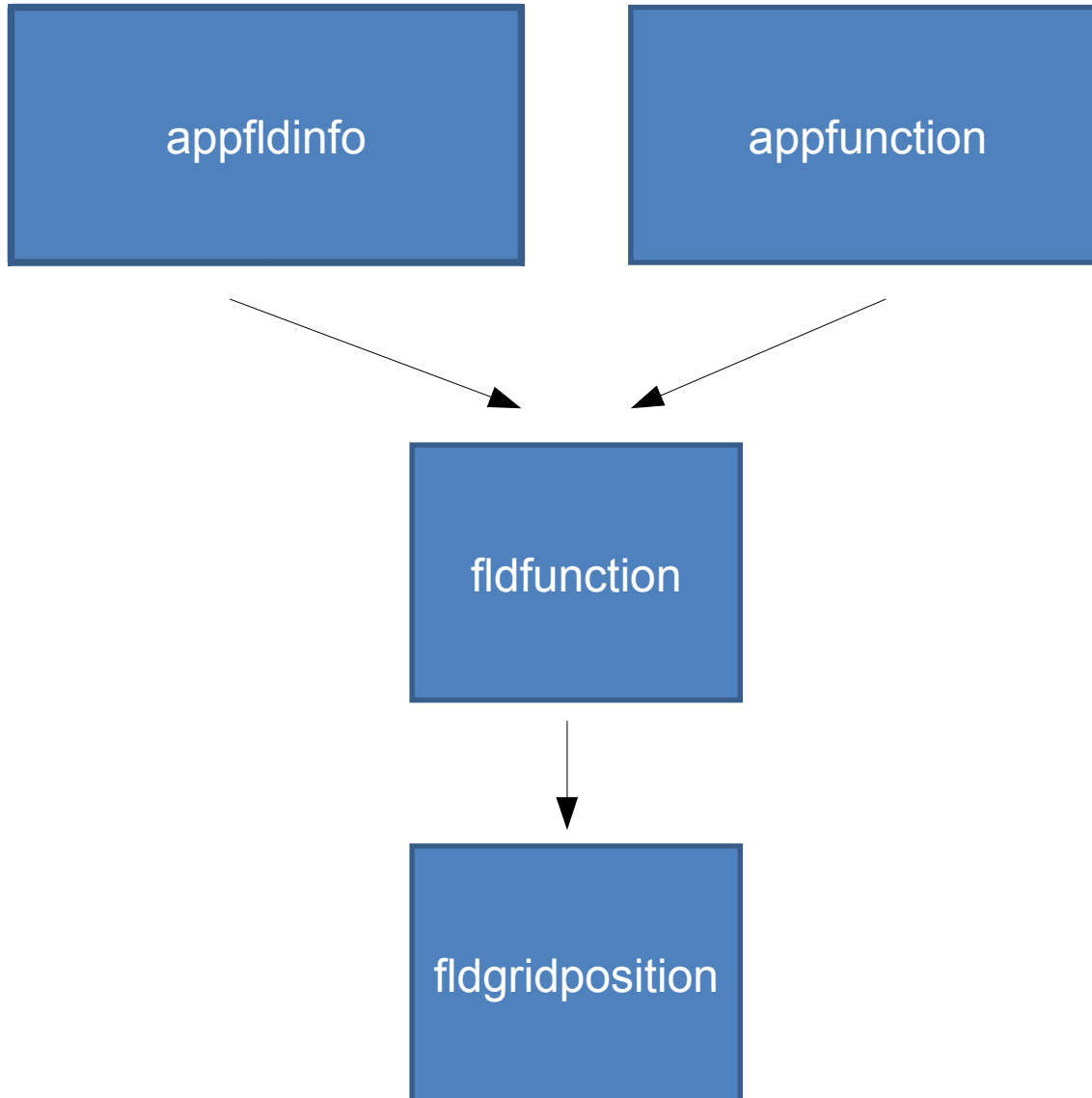
Structure of Meta Info (1)



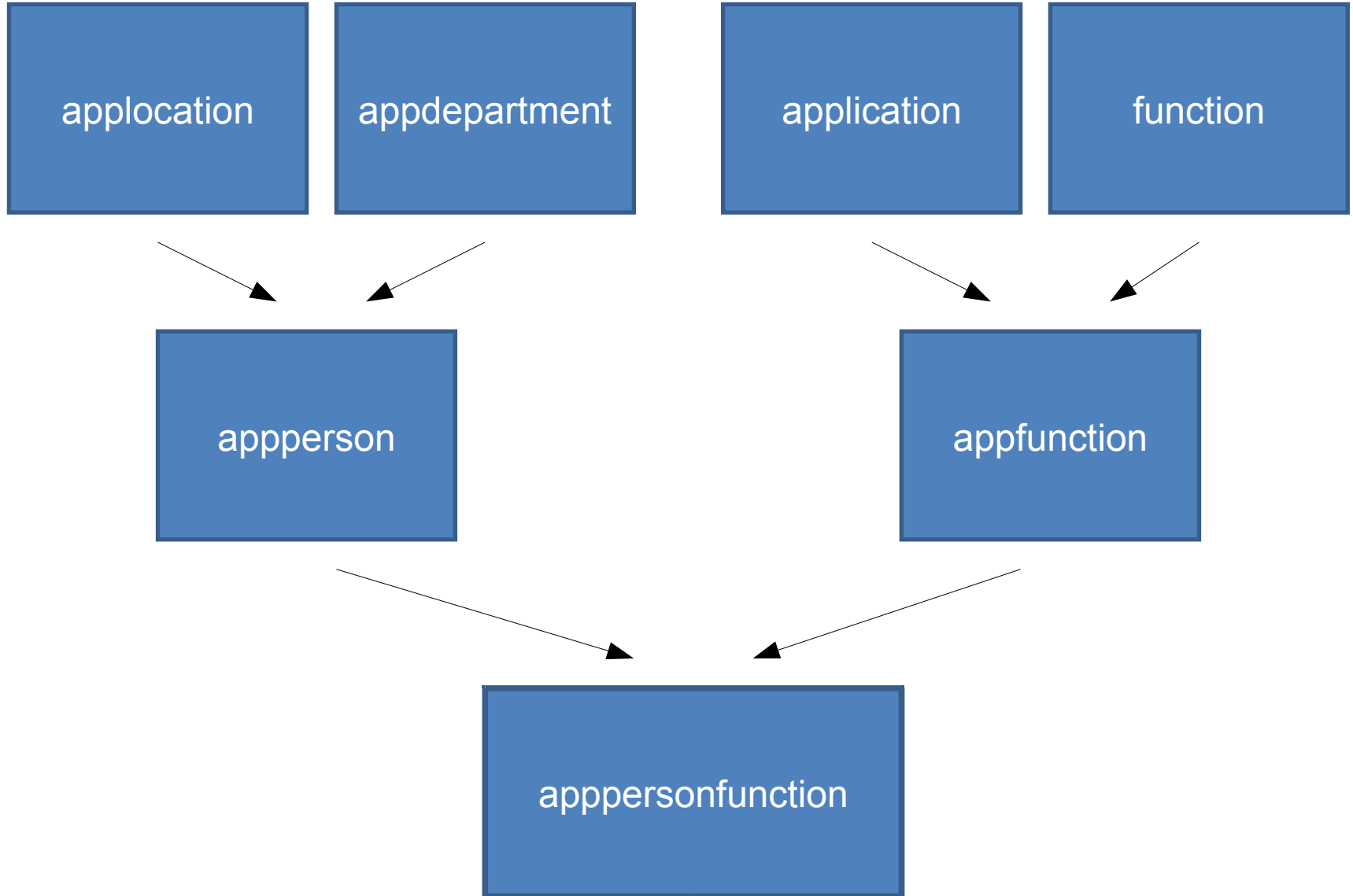
Structure of Meta Info (2)



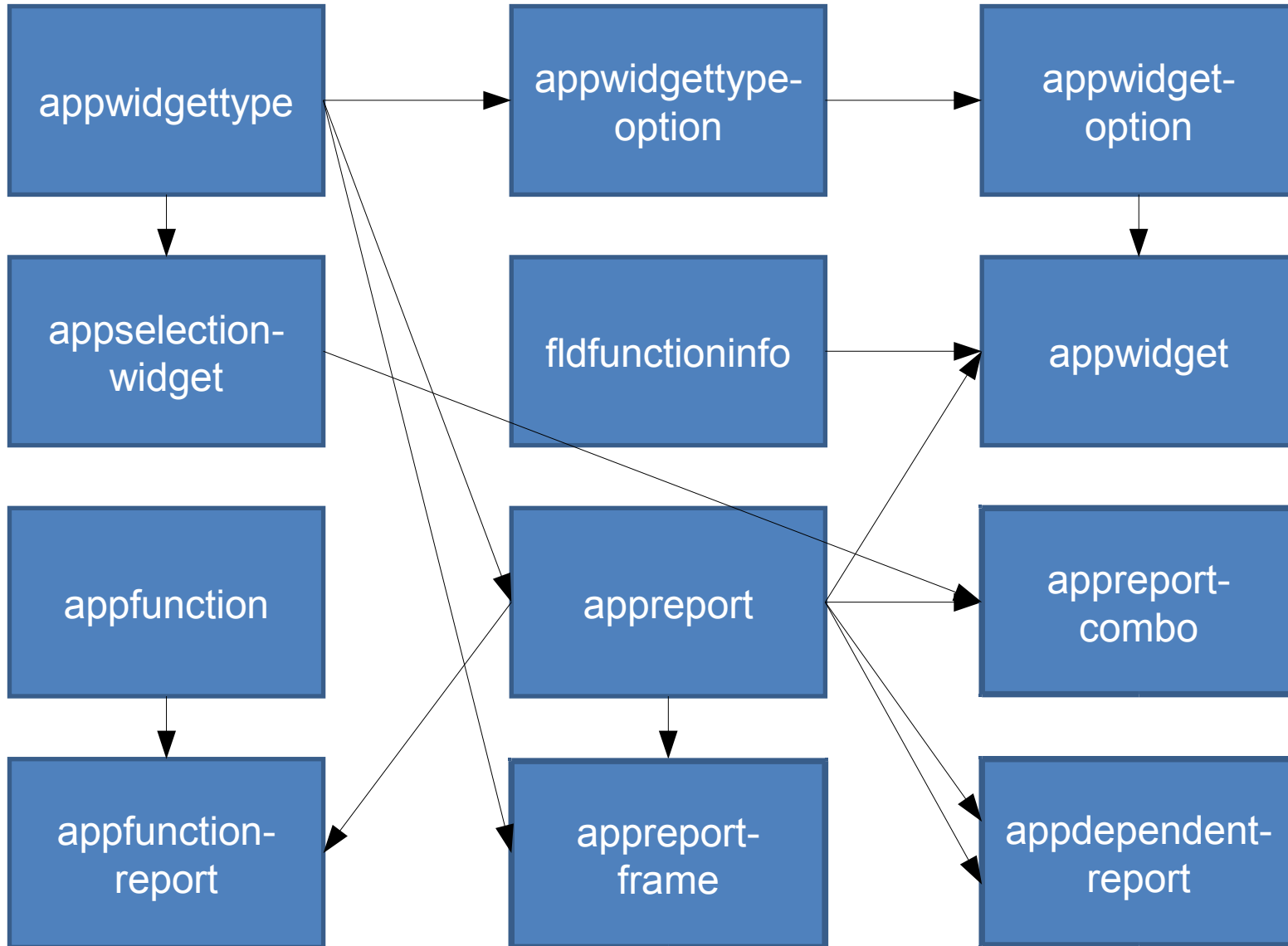
Structure of Meta Info (3)



Structure of Meta Info (2)



Report Meta Info



Building an InputReport

- Filling of the field meta information tables
- Filling of the report meta information
- Running the application

Status

- This project is a work in progress
- ATM not much activity because of other projects
- Possibly influenced in the future from ATWF
- Will be continued!

Conclusions (1)

- As tdbc is now available use of tdbc information instead of apptabname, appfldname, appfldattr tables
- Clean up/redesign of report meta information
- The information passed to the client should be in a more generic format like for example dicts
- Should eventually try to use as a second UI Aejaks

Conclusions (2)

- Project started to make ideas from payed work an open source project
- Should allow fast development of some dedicated class of applications (no general purpose tool)
- Use of newer technology allows easier maintainance of code base
- Additional developpers would speed up implementation
- No web frame work, but a tool for building client/server applications, which can be started via an URL from a browser