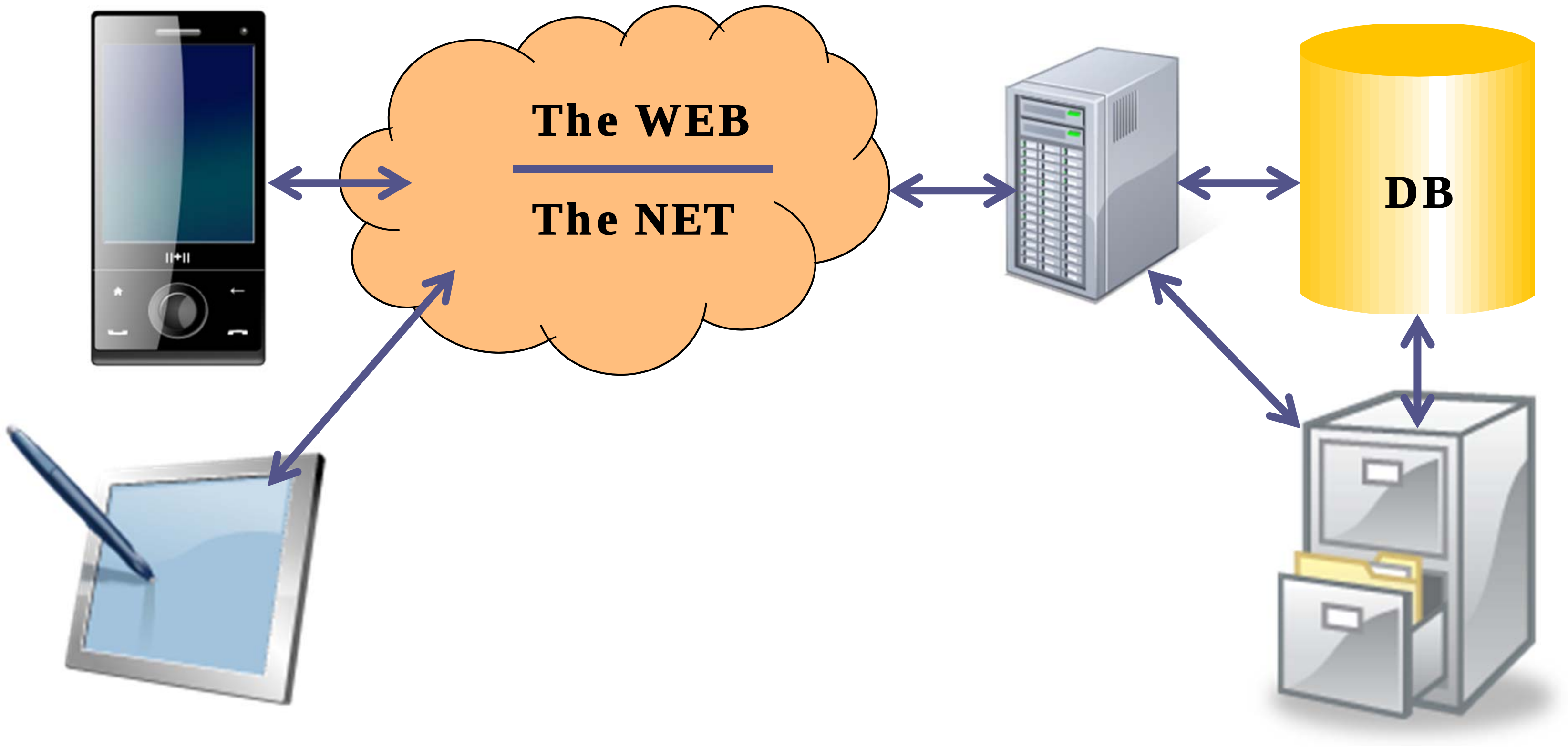
What is Computer Science?

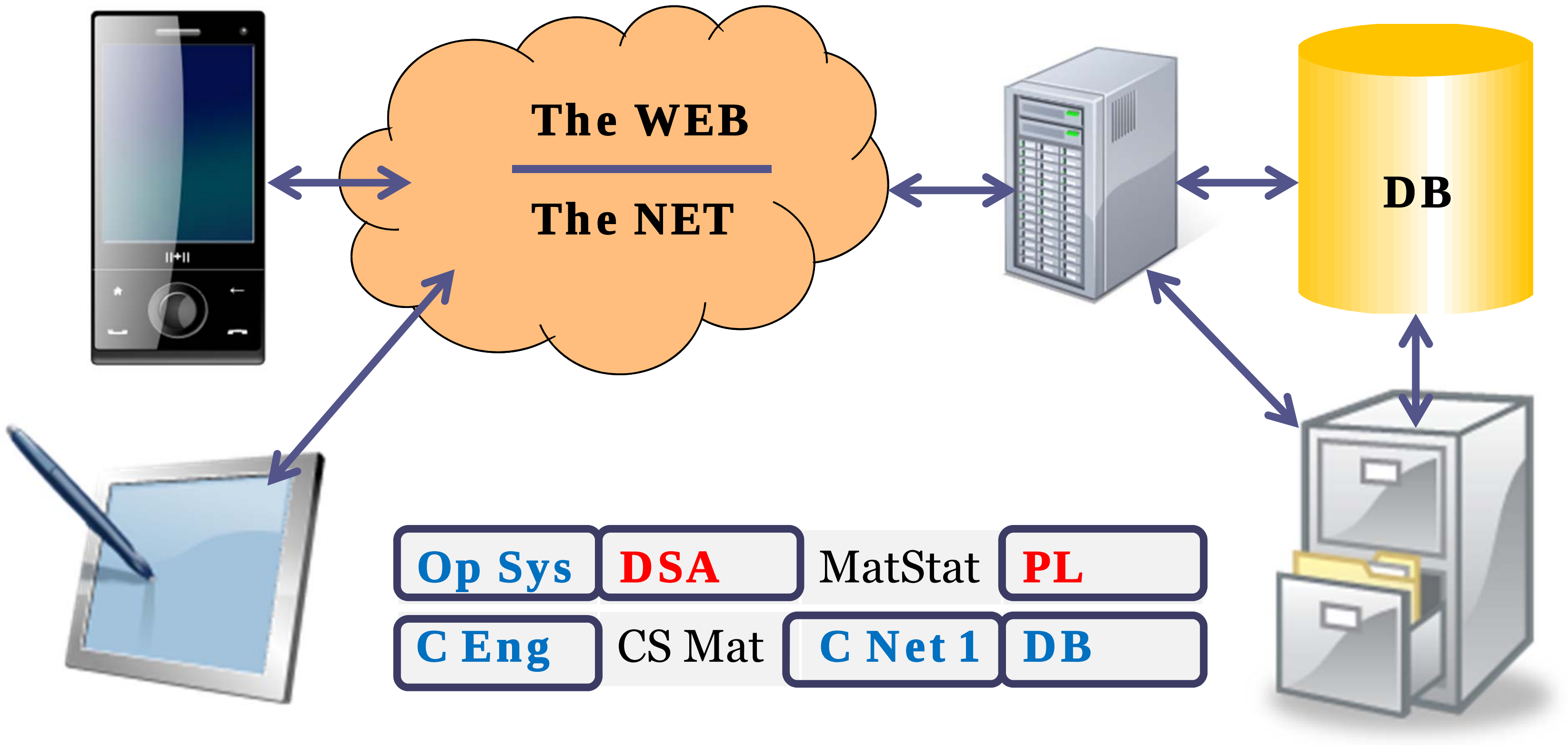
A short overview of our courses
and existing computer systems

Donald F. Ross
Office: 21D 413
Email: donald.ross@kau.se

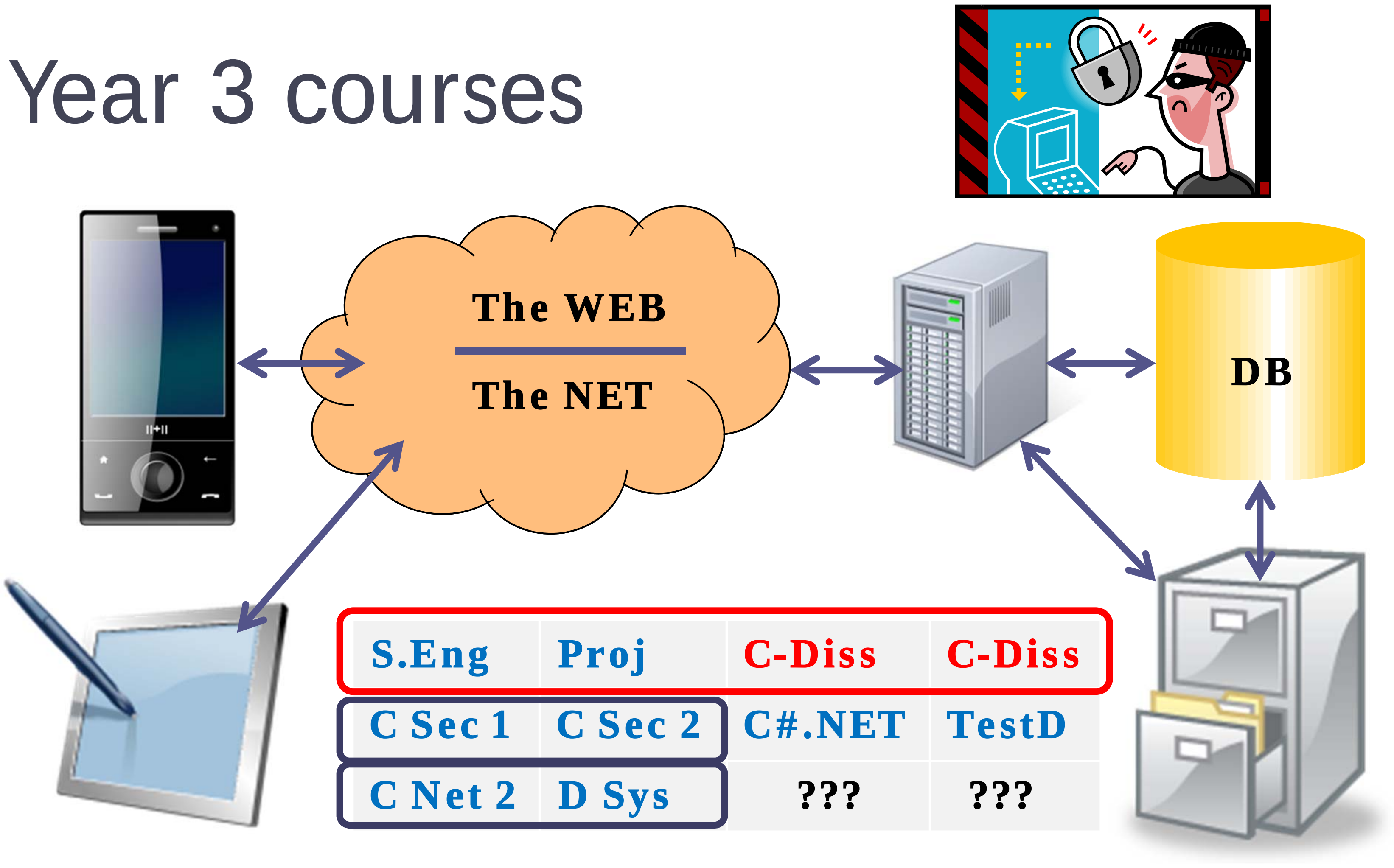
Existing systems



Existing systems + year 2 courses

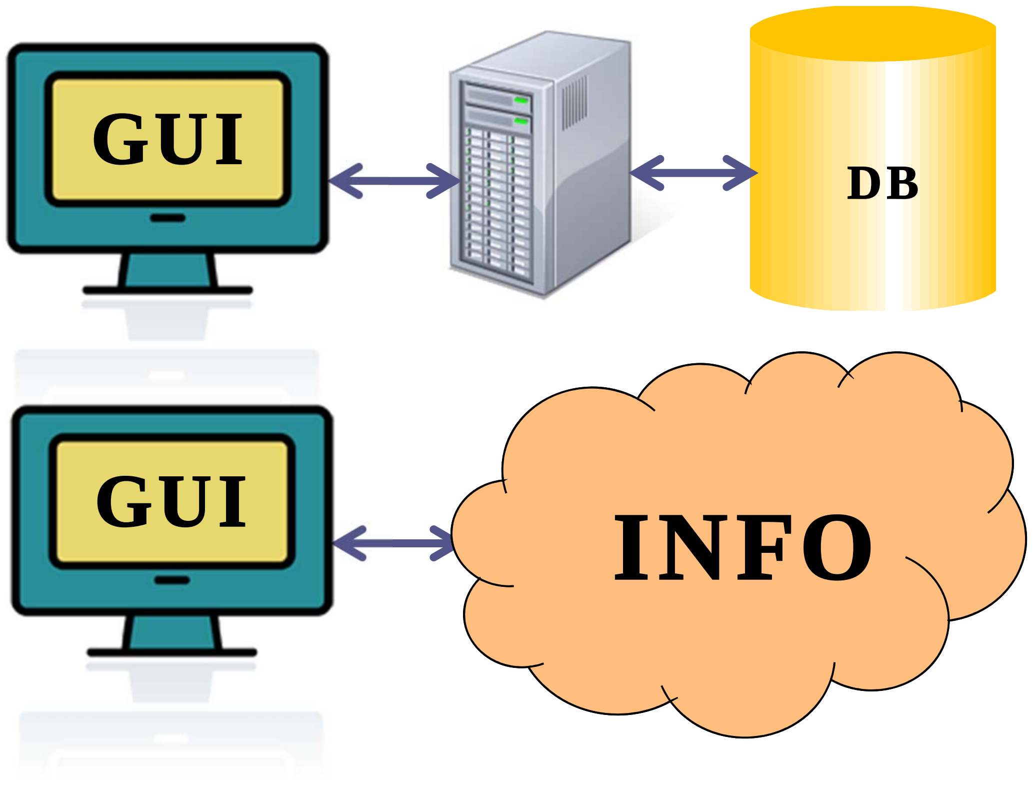


Year 3 courses

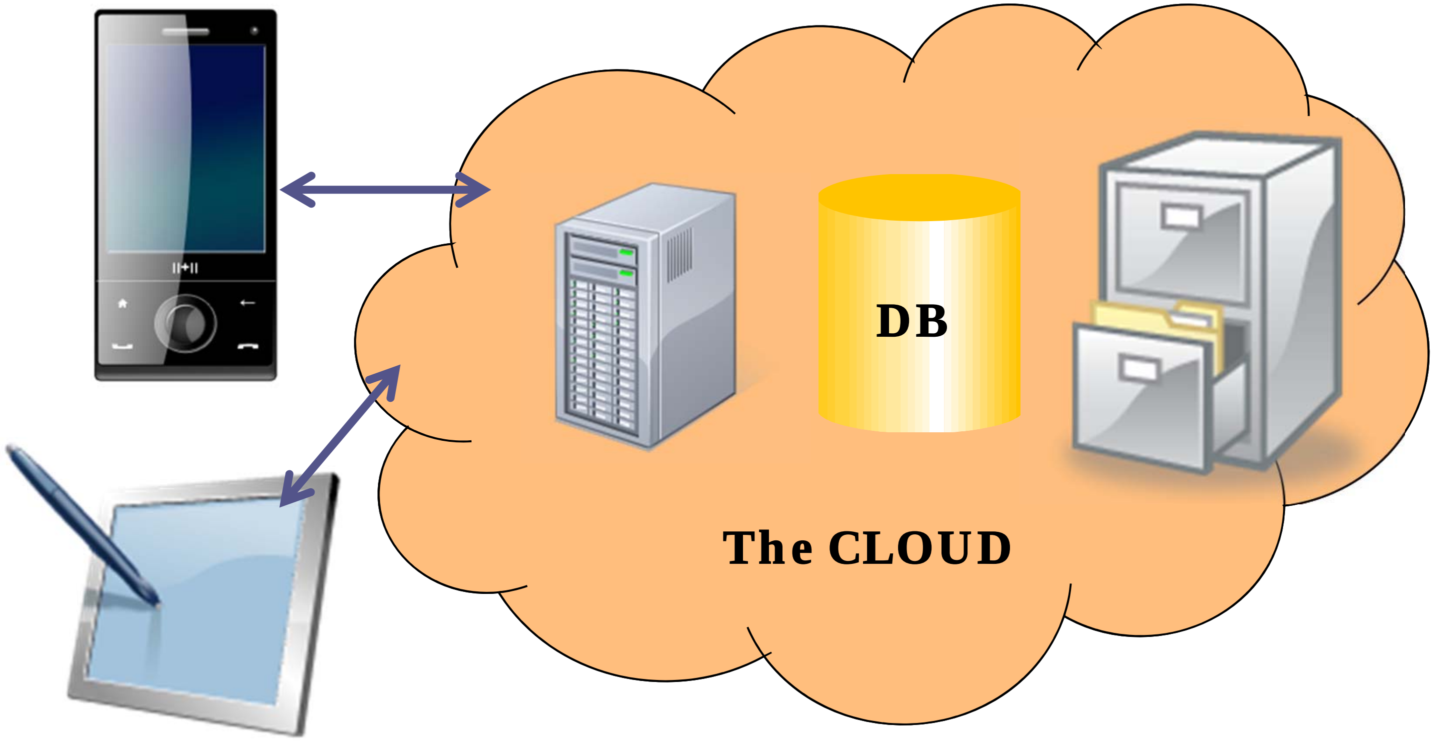


S.Eng	Proj	C-Diss	C-Diss
C Sec 1	C Sec 2	C#.NET	TestD
C Net 2	D Sys	???	???

Existing systems: abstraction

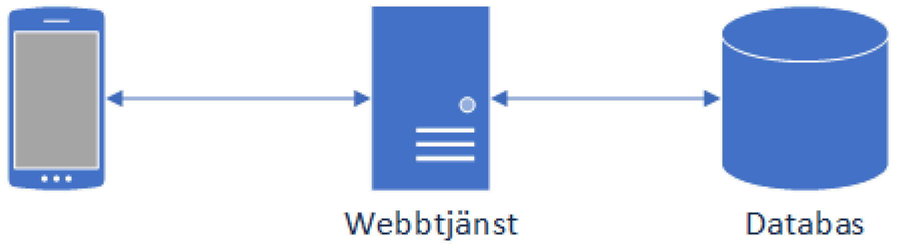


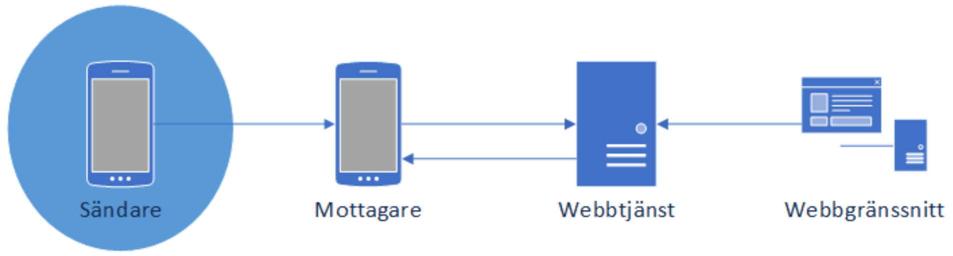
Future systems



iBeacon – context based information

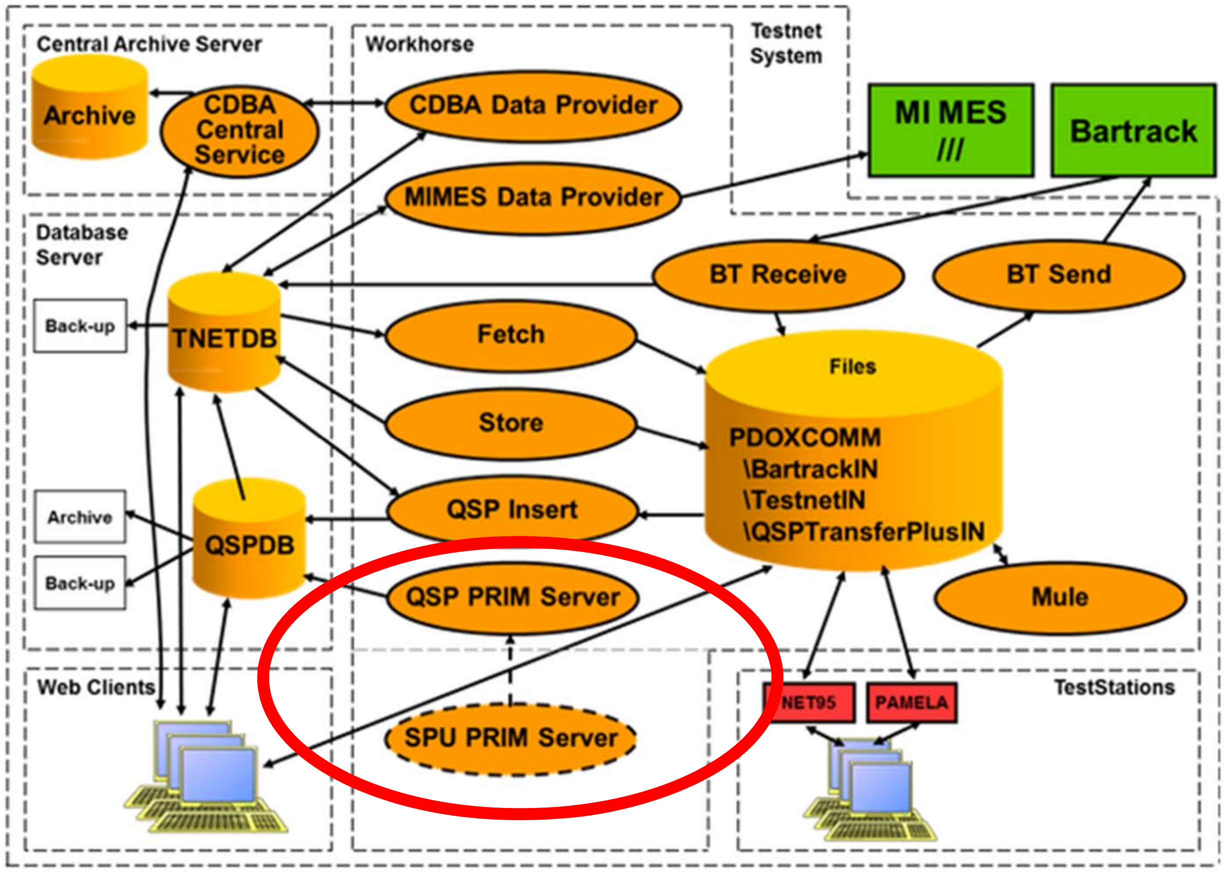
C-dissertation Lönnerstrand / Älveborn Spring 2014





TestNet System – Prevas / Eriksson

C-Dissertation Johnsson/Ljungdahl Spring 2014



An Aside: On learning

A school in England - motivation & learning goals:

1. **Grit**
   - **or wellbeing, the ability to bounce back**
2. **Professionalism**
   - **to know what it takes to do something well**
3. **Expertise**
   - **to gain the knowledge and ability to think in a range of disciplines**
4. **Eloquence**
   - **speaking and thinking in a sophisticated way**
5. **Spark**
   - **the ability to think laterally and generate new ideas**
6. **Craftsmanship**
   - **the techniques for redrafting and improving work until it is beautiful**

Thank you!

- Questions?

- Now on to DSA…
  - … a comment on programming style
  - In DSA I will also prepare you for the Programming Languages course…
  - … there we will look at different PROGRAMMING PARADIGMS – functional and logic

Cross paradigm influences
The conditional expression in C / functional PLs

"Functional C"

```
static listref be_add_val(valtype v, listref L)
{
 return is_empty(L)           ?  create_e(v)
  :  v < get_value(head(L)) ?  cons(create_e(v), L)
  :                            cons(head(L), be_add_val(v, tail(L)));
}
```

LISP

```
 (defun b_add (v L)
   ( cond
     ( (null L)                (list v))
     ( (< v (first L))         (cons v L))
     ( t                       (cons (first L)  (b_add v (rest L)))
     )
   )
;; NB: cond, null, list, first, cons, rest are built-in in LISP
```

Cross paradigm influences
The conditional expression in C / functional PLs

"Functional C"

```
static listref be_add_val(valtype v , listref L)
{
 return is_empty(L)            ?  create_e(v)
   :  v < get_value(head(L)) ?  cons(create_e(v), L)
   :                            cons(head(L), be_add_val(v, tail(L)));
}
```

Haskell (simpler!)

```
 bAdd v [ ]                 = v: [ ]
 bAdd v [x:xs]
   | v < x                  = v : [x:xs]
   | otherwise              = x : bAdd v xs
```

':' is the cons operation