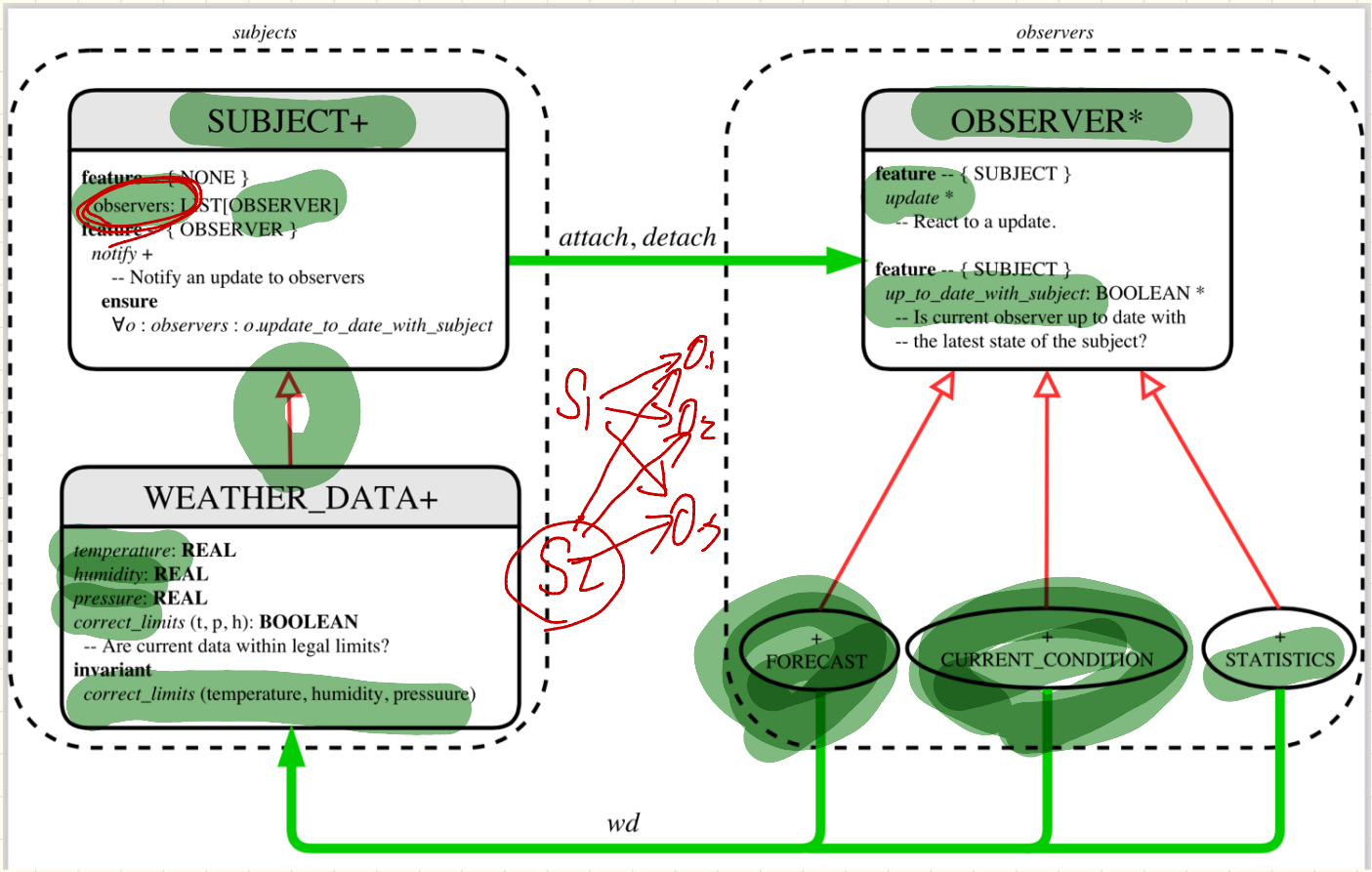
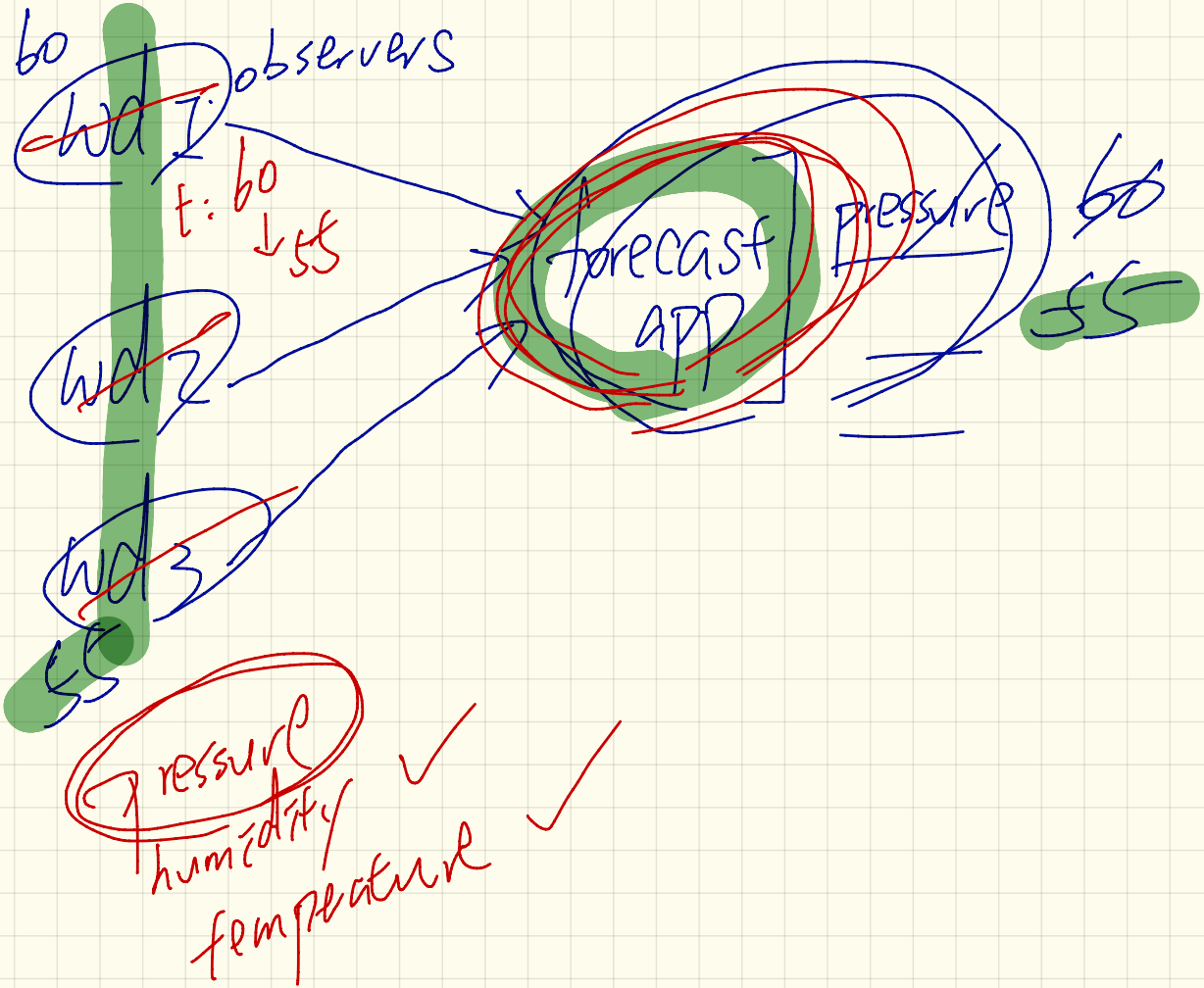
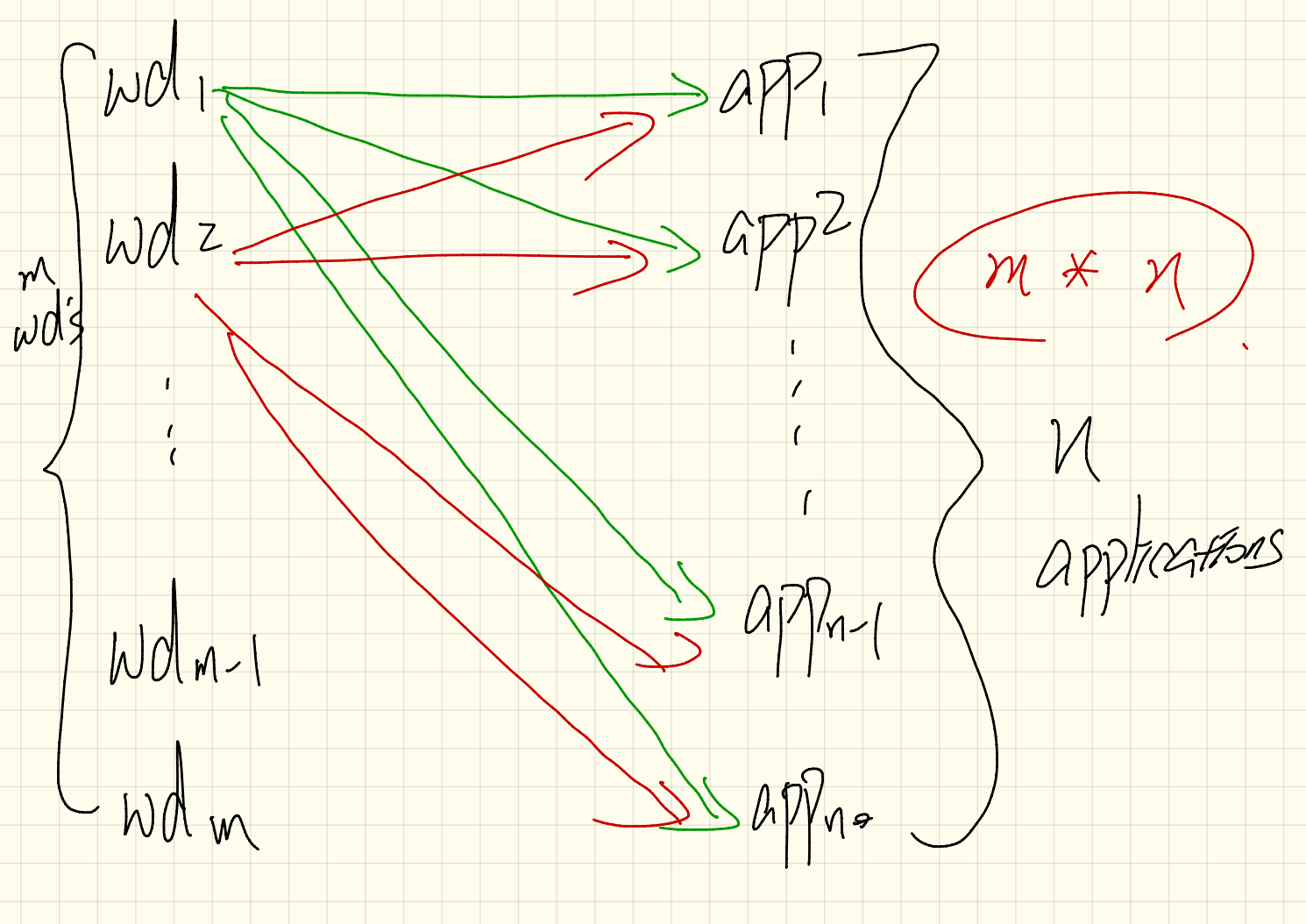
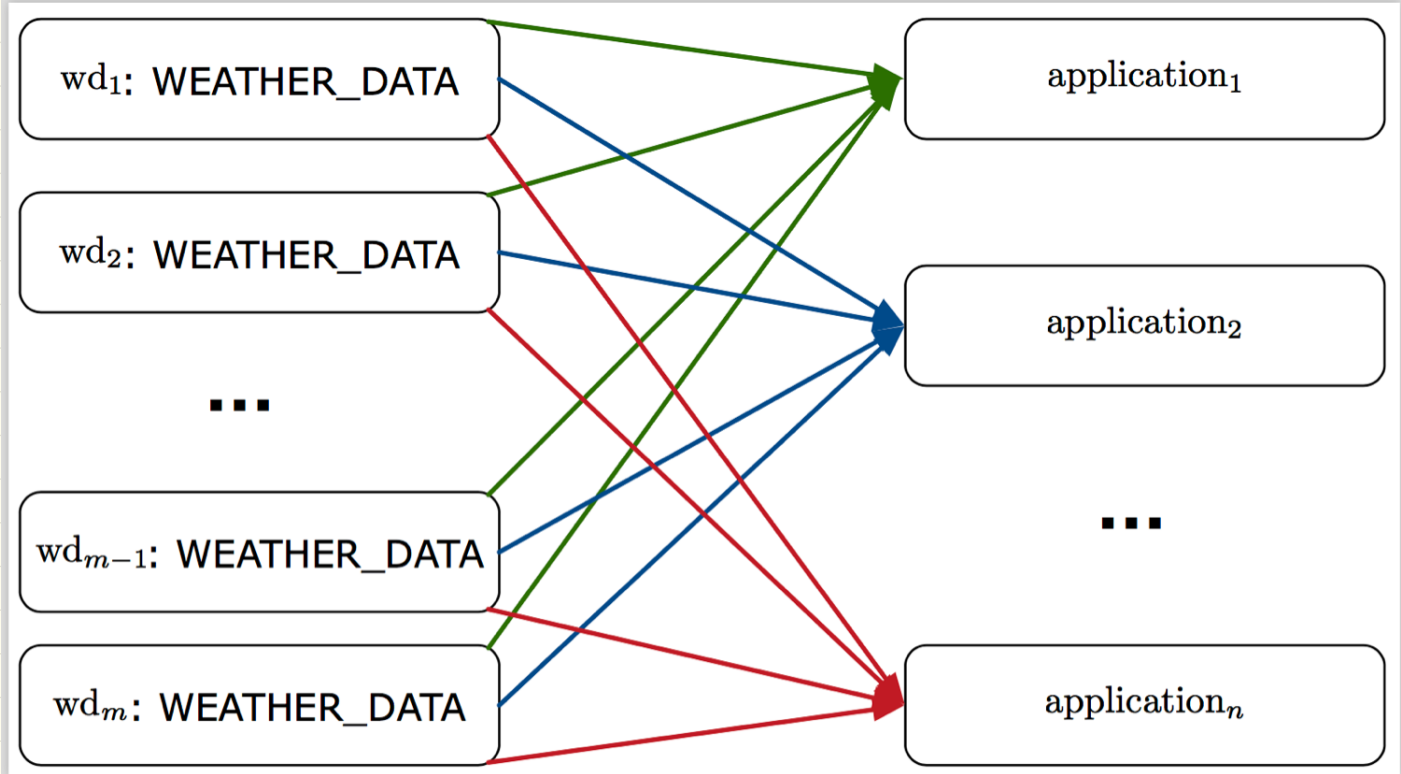
Lecture 19

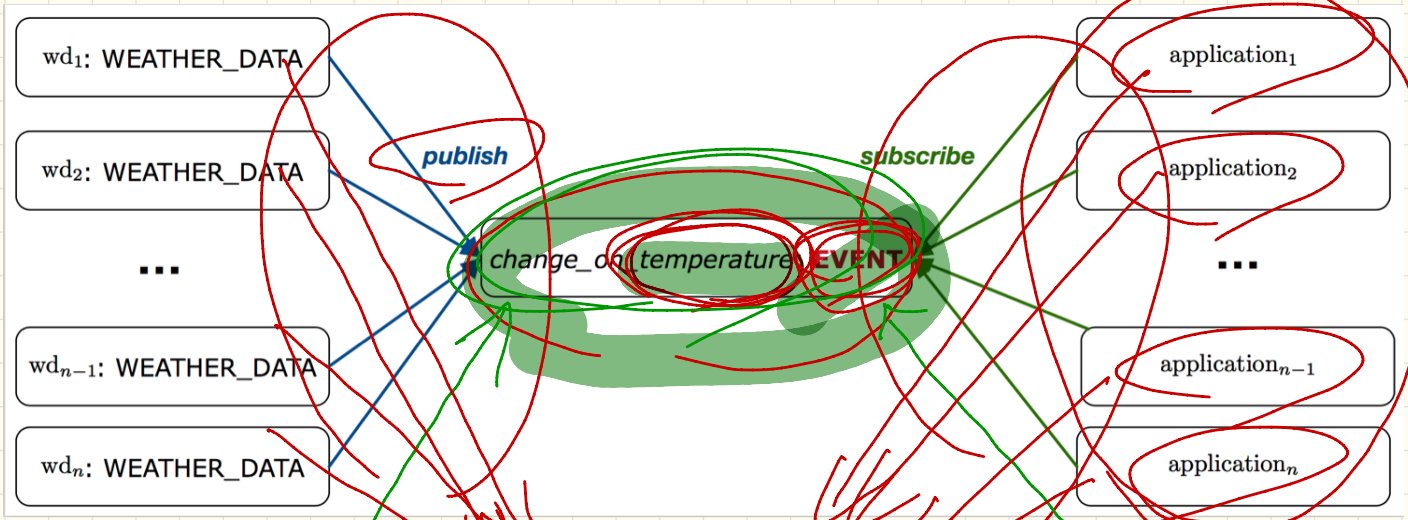
Wednesday Nov. 15

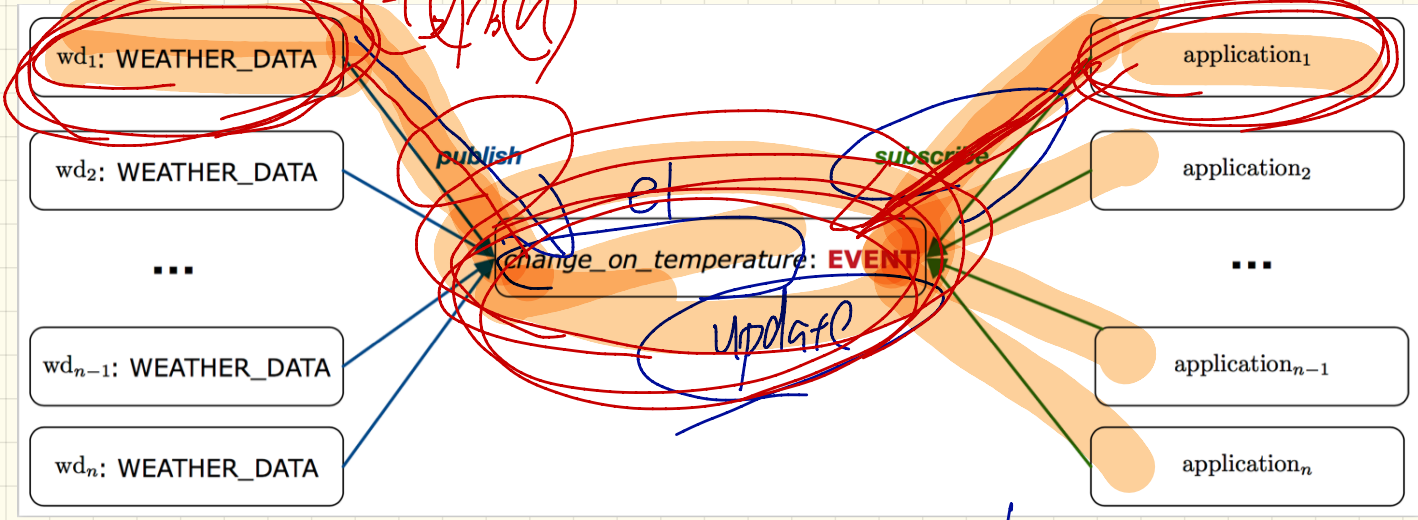

subjects
observers
SUBJECT+
feature -- { NONE }
observers: LIST[OBSERVER]
feature -- { OBSERVER }
notify +
-- Notify an update to observers
ensure
∀o : observers : o.update_to_date_with_subject
attach, detach
OBSERVER*
feature -- { SUBJECT }
update *
-- React to a update.
feature -- { SUBJECT }
up_to_date_with_subject: BOOLEAN *
-- Is current observer up to date with
-- the latest state of the subject?
WEATHER_DATA+
temperature: REAL
humidity: REAL
pressure: REAL
correct_limits (t, p, h): BOOLEAN
-- Are current data within legal limits?
invariant
correct_limits (temperature, humidity, pressuure)
S1
S2
O1
O2
O3
+
FORECAST
+
CURRENT_CONDITION
+
STATISTICS
wd


bo
observers
wd 1
wd 2
wd 3
t: bo
ss
forecast
app
pressure
bo
ss
pressure
humidity
temperature


wd1
wd2
wdm-1
wdm
m wd's
app1
app2
appn-1
appn
n applications
m * n

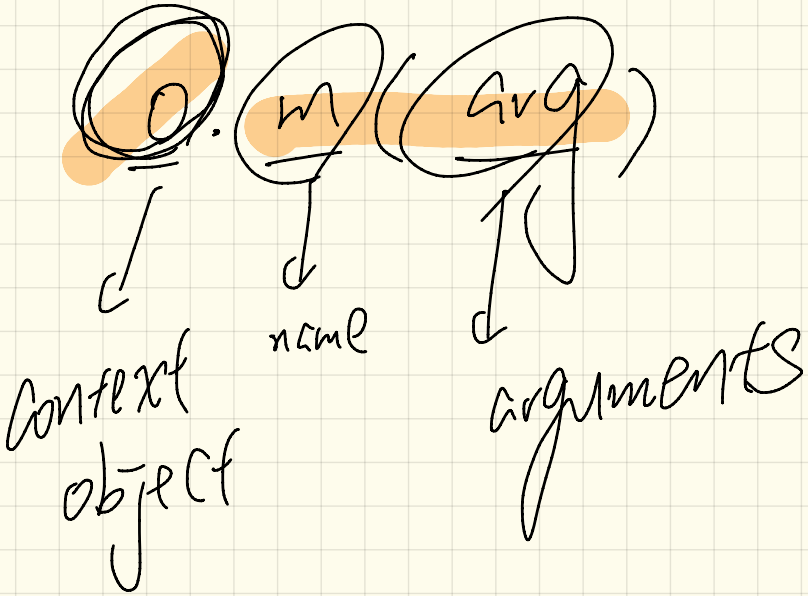

wd$_1$: WEATHER_DATA
wd$_2$: WEATHER_DATA
...
wd$_{m-1}$: WEATHER_DATA
wd$_m$: WEATHER_DATA
application$_1$
application$_2$
...
application$_n$


wd1: WEATHER_DATA
wd2: WEATHER_DATA
...
wdn−1: WEATHER_DATA
wdn: WEATHER_DATA
publish
change_on(temperature):EVENT
subscribe
application1
application2
...
applicationn−1
applicationn

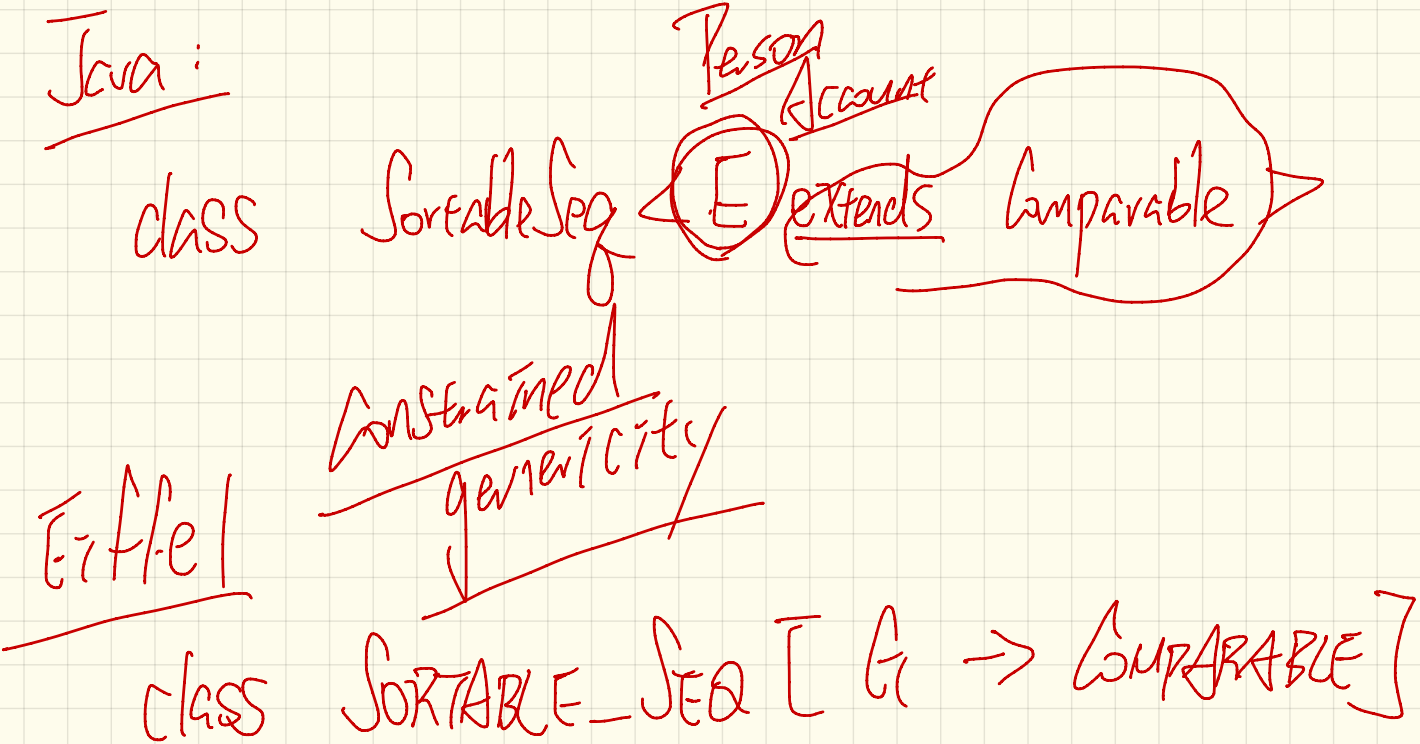


app1 subscribe to e1 (attach pointer to some update function)

When there's an update from wd1, it will publish to e1, calling the saved update op.



listener.action(args);


Java :
class SortableSeq < E extends Comparable
Person
Account
Constrained genericity
Eiffel
class SORTABLE_SEQ [G -> COMPARABLE]

t_1: TUPLE

t_2: TUPLE [REAL]

t_3: TUPLE [REAL; REAL]

```
class Point {
  int x;
  void change_x(int nx) {
    x := nx;
  }
}
```

this.x