

Linq:

— YIELD RETURN;  
    ↳ ITERATOR

— EXTENSION METHODS

- WHERE
- SELECT

CODE  
---  
RAISE EVENT1  
---  
RAISE EVENT2  
---  
RAISE EVENT3

CLIENT.  
ABON EVENT1  
ABON EVENT2

RAISE EVENT1  
---

RAISE EVENT2  
---

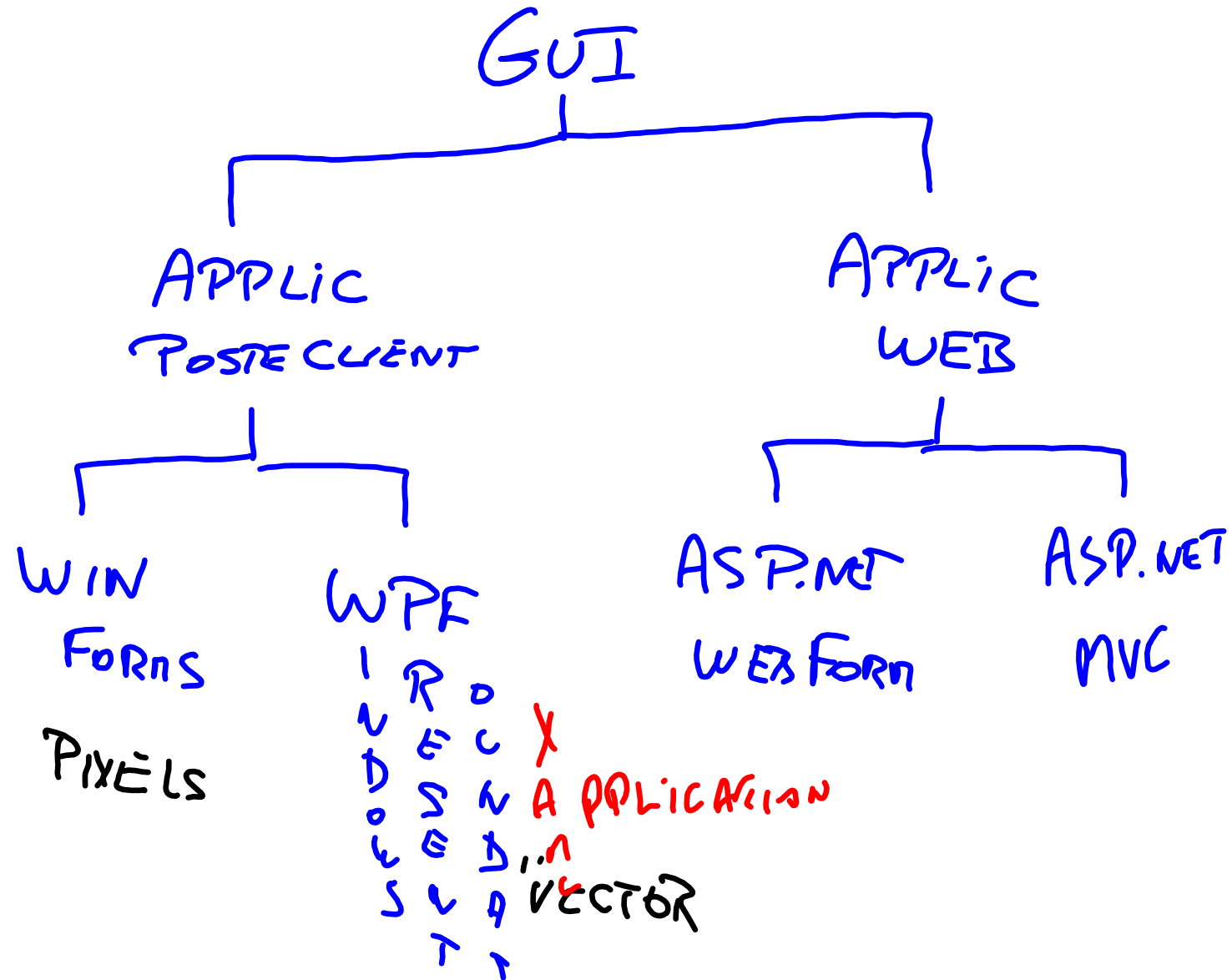
Method PARTIAL

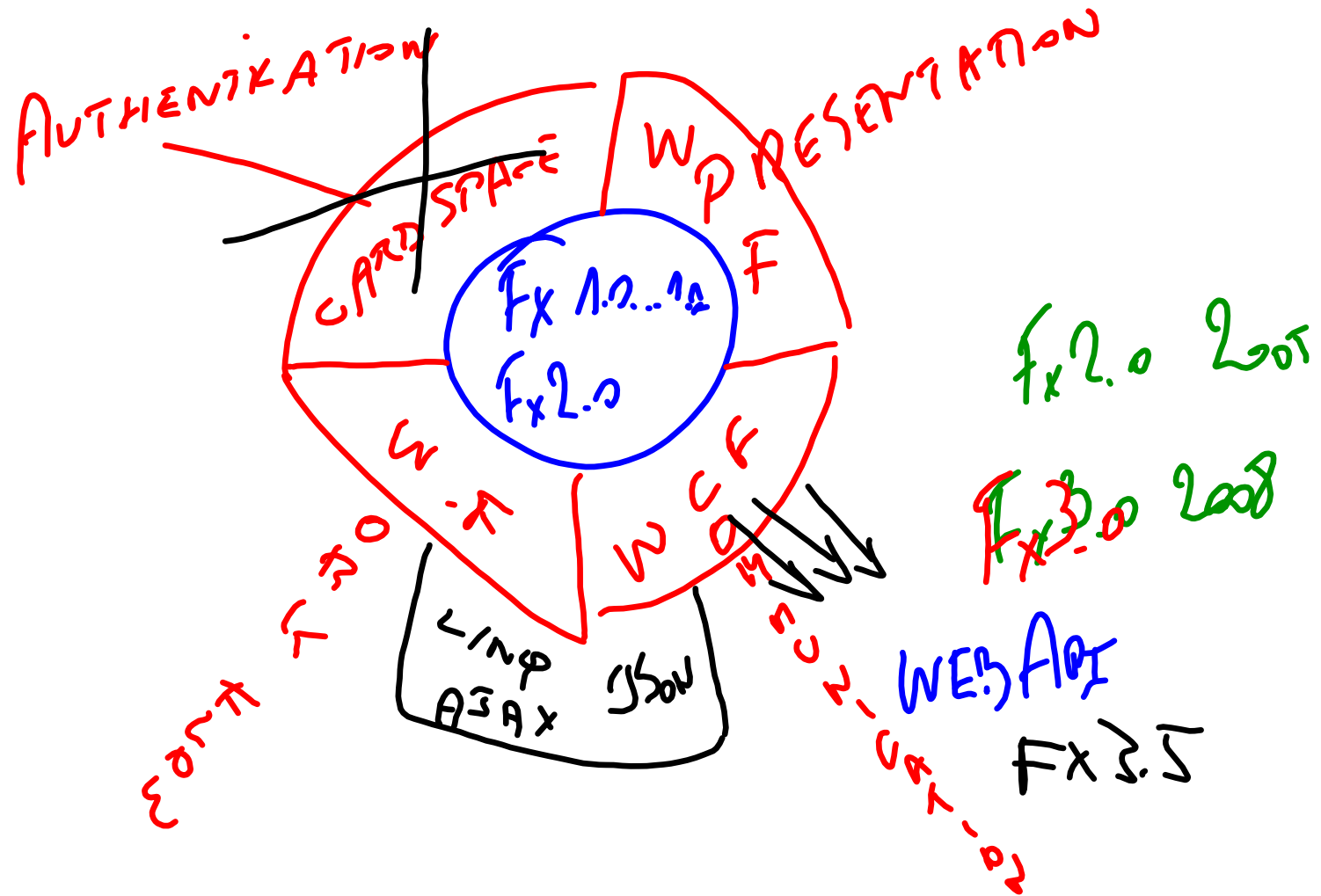
CODE

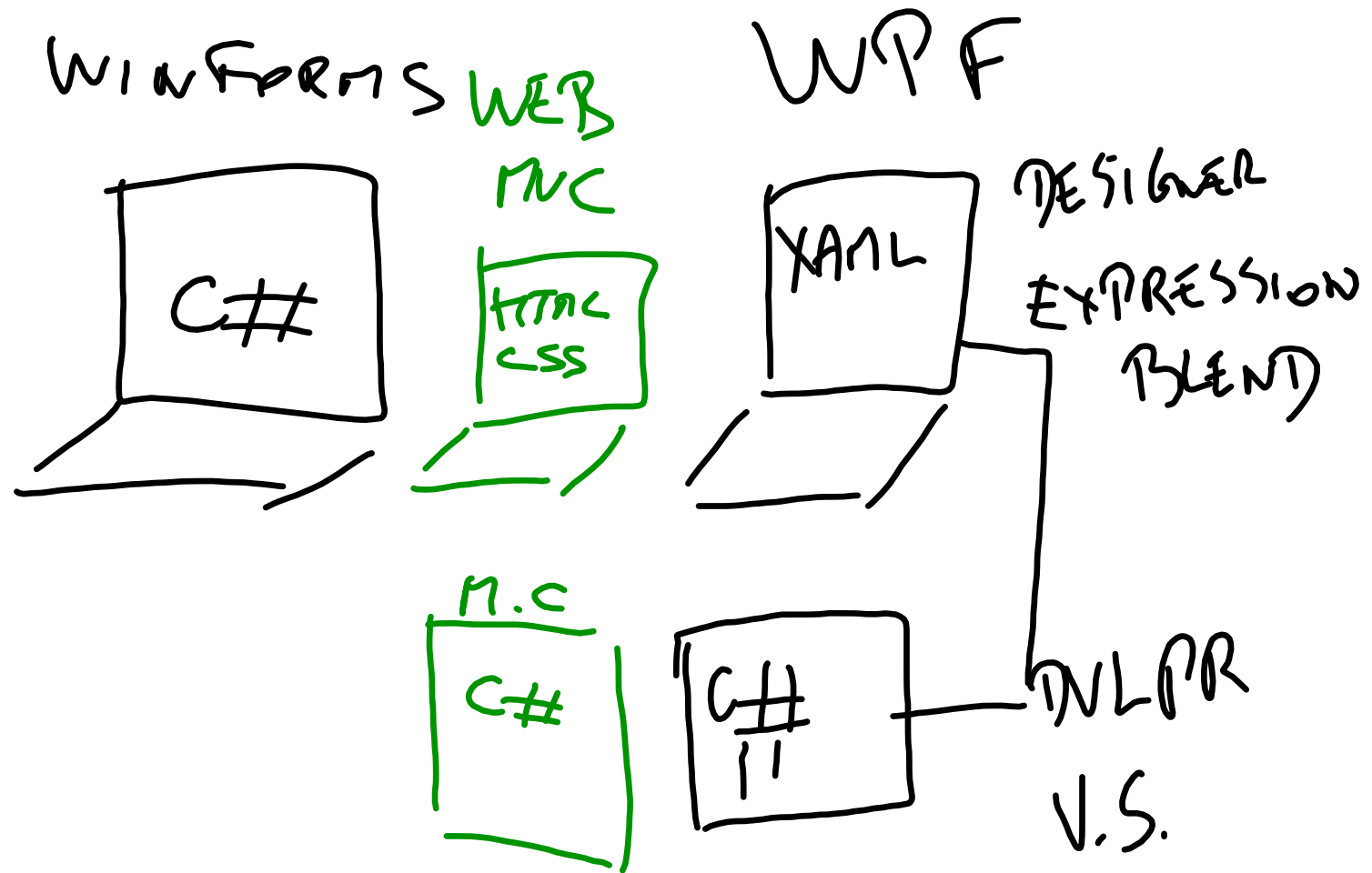
—  
—  
—  
—  
MP1( )  
—  
—  
—  
—  
MP2( )  
—  
—  
—  
—  
MP3( )

CLIENT

MP1( )  
—  
—  
—  
—  
MP2( )  
—  
—  
—  
—  
—







VB ← 2009

ASP  
CEA  
TRG  
VE  
ER  
R

JSP

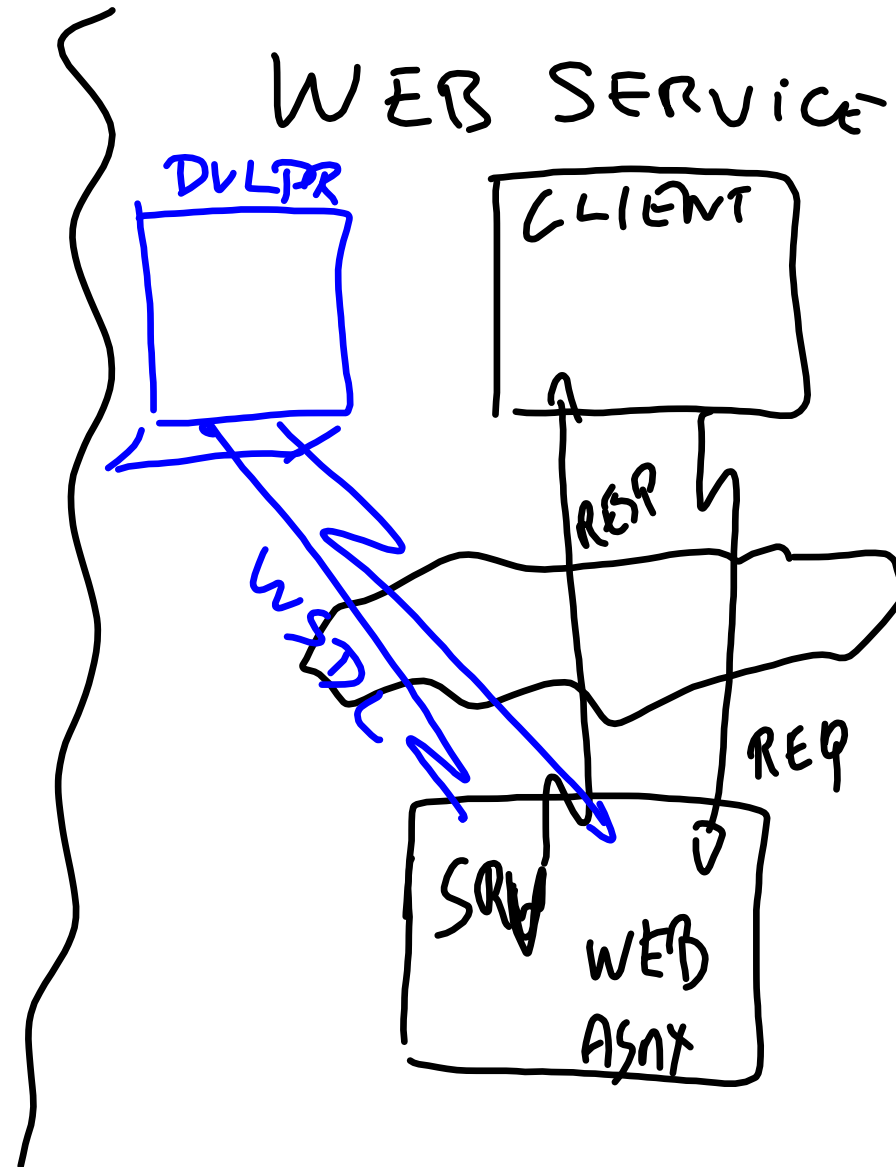
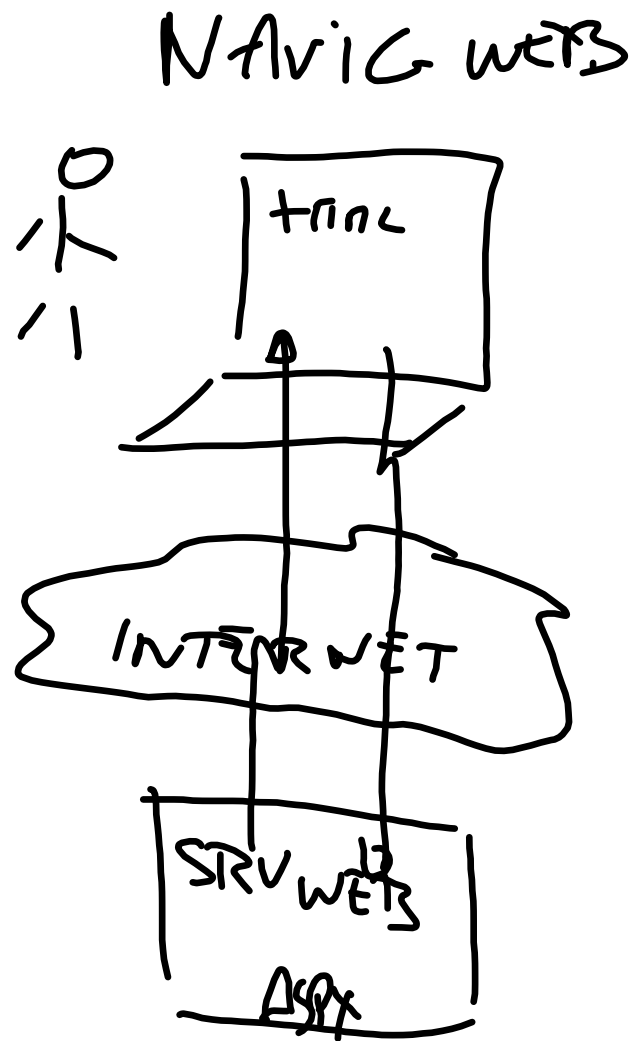
.HTM  
STATIC  
ASP+  
ASPX

WS  
AUTHENTICAT

ASMX

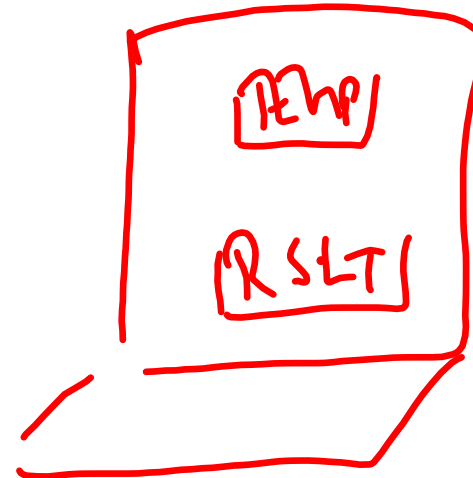
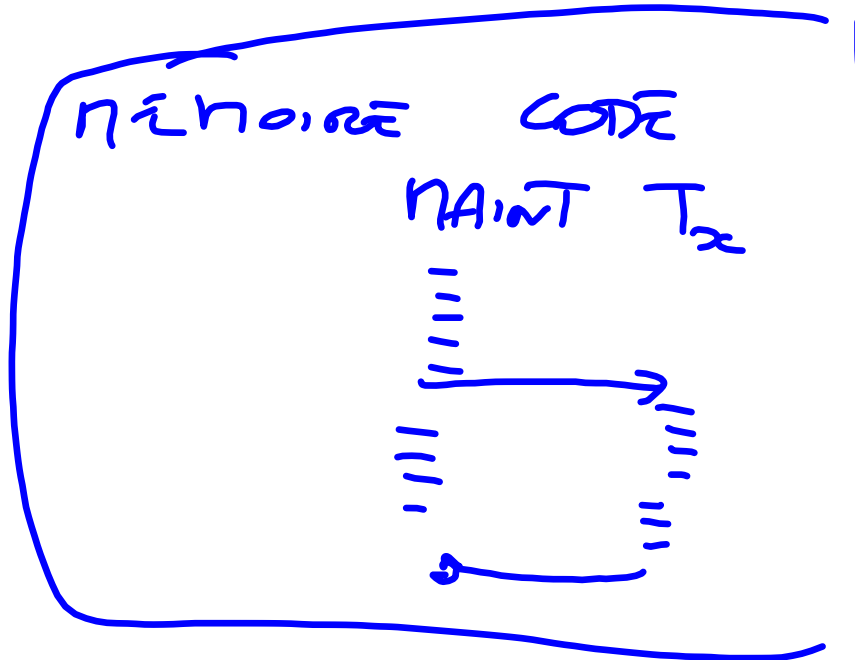
CEE  
TAT  
iUH  
UEO  
FRD

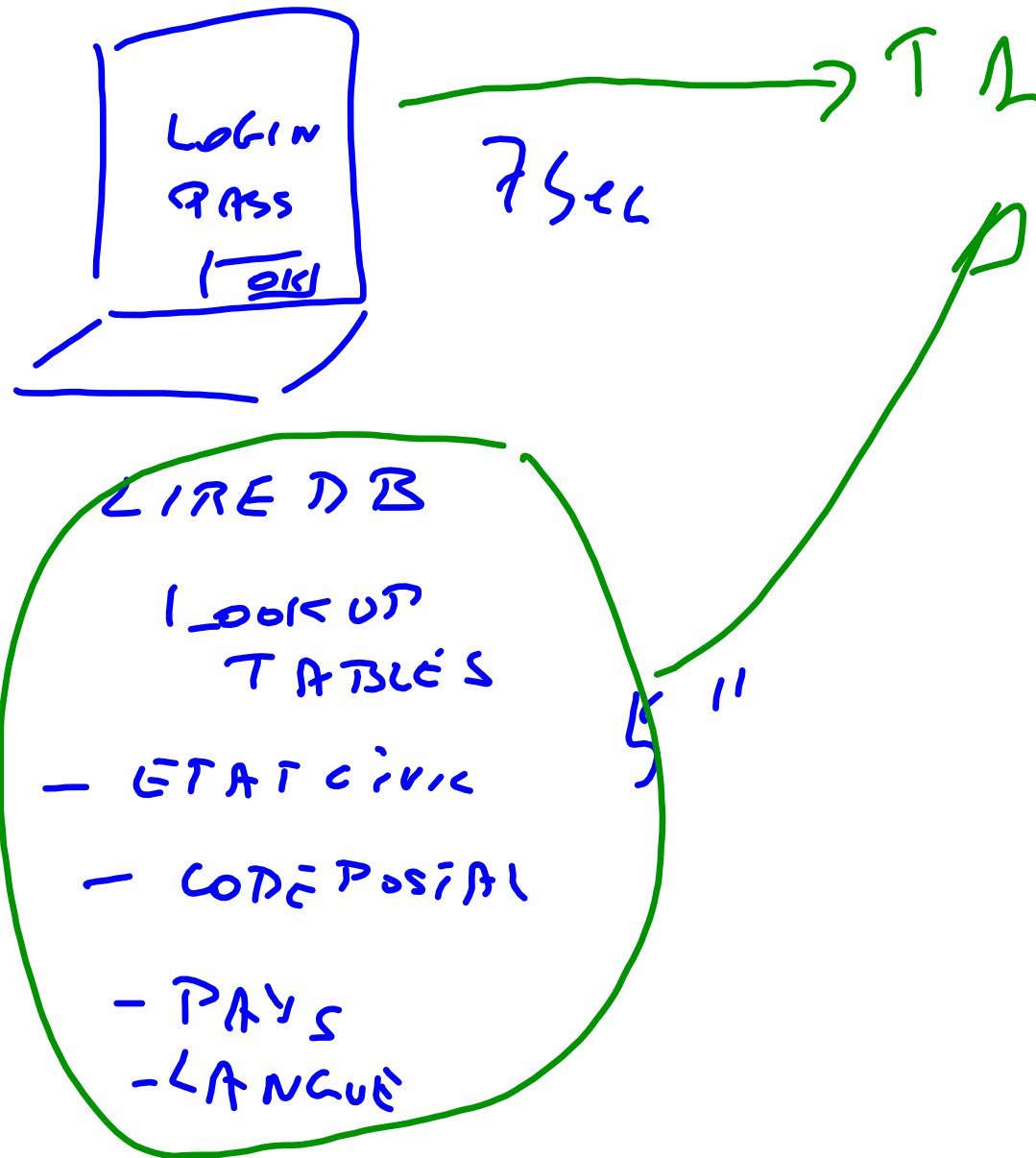
UUDI

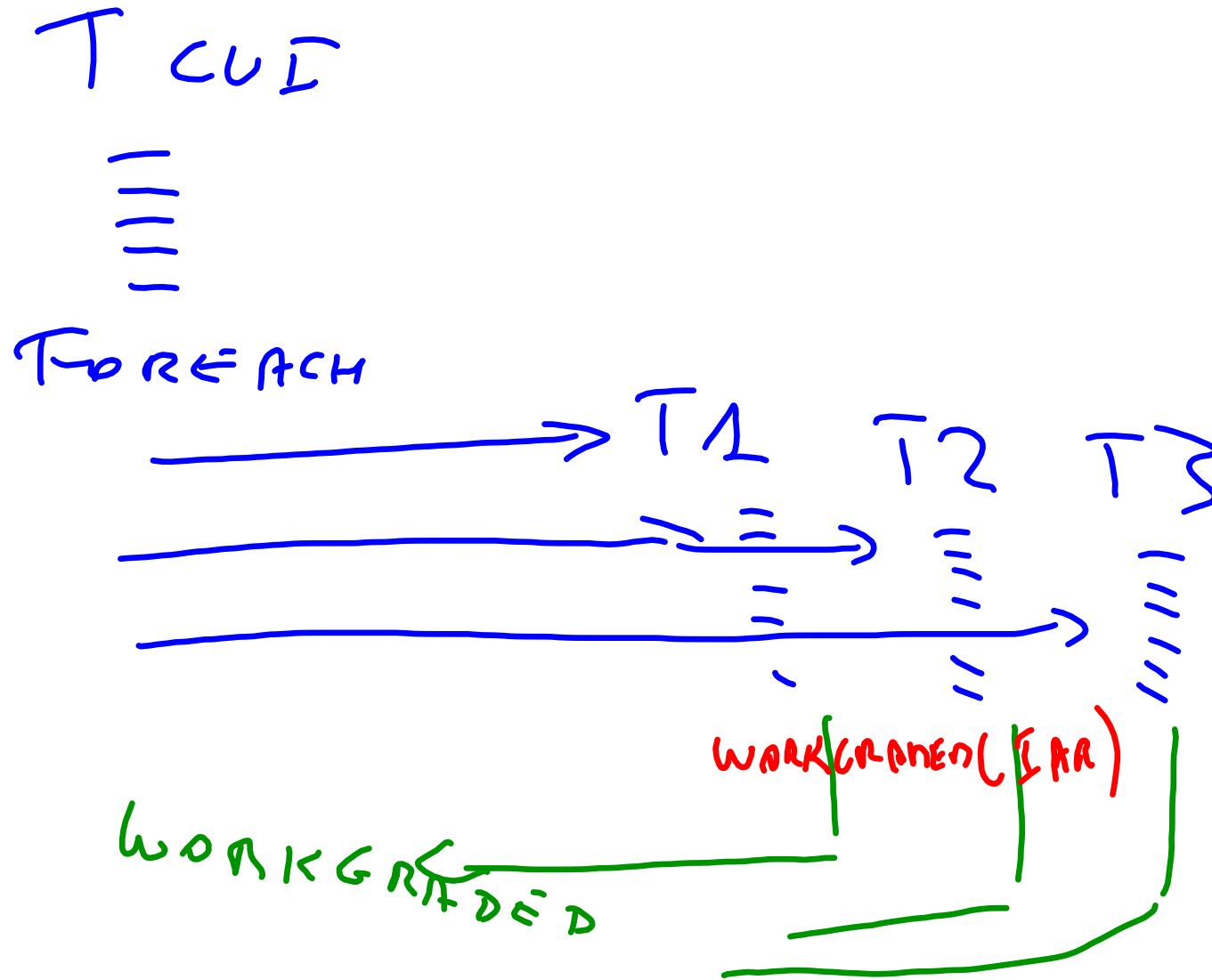


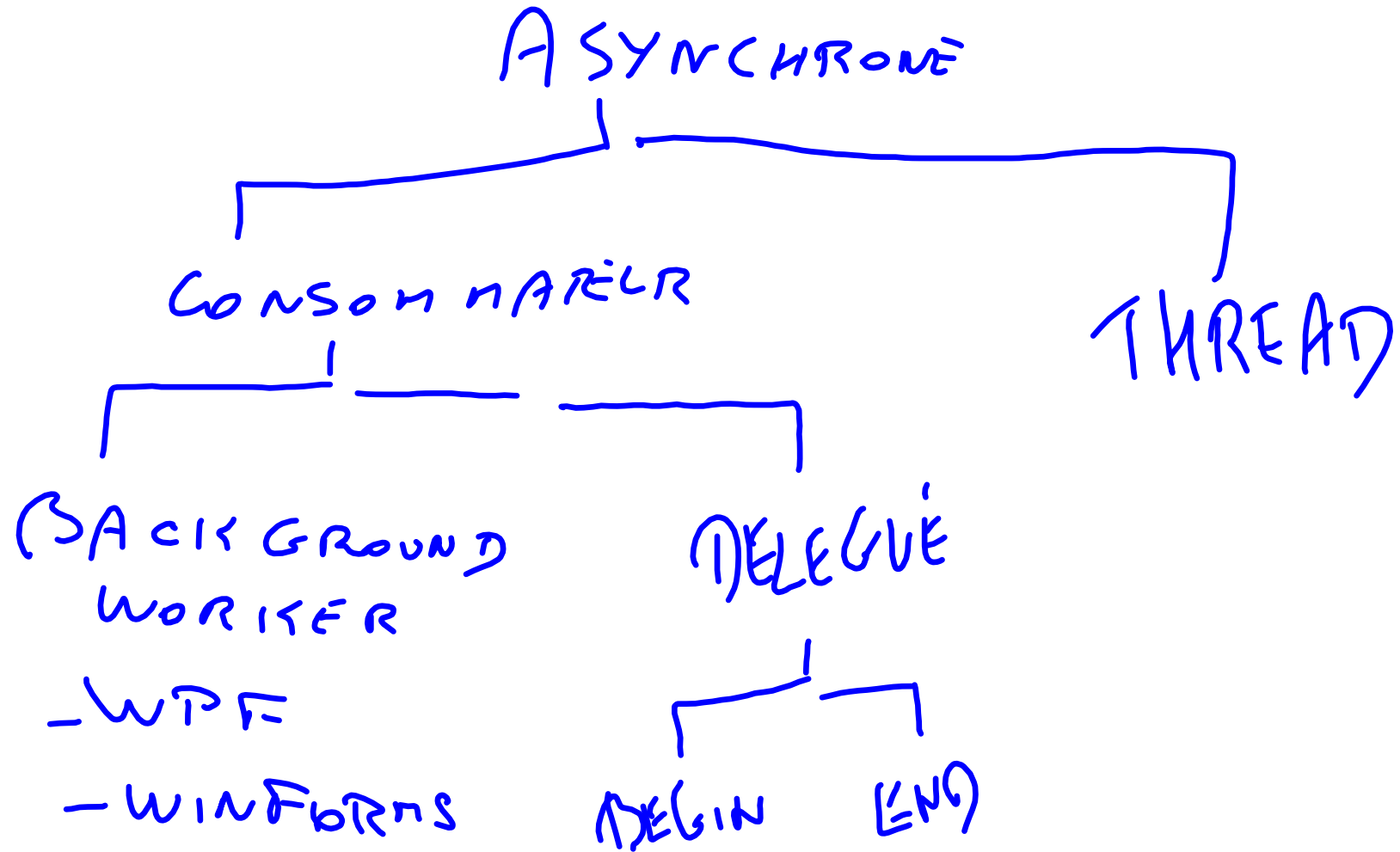


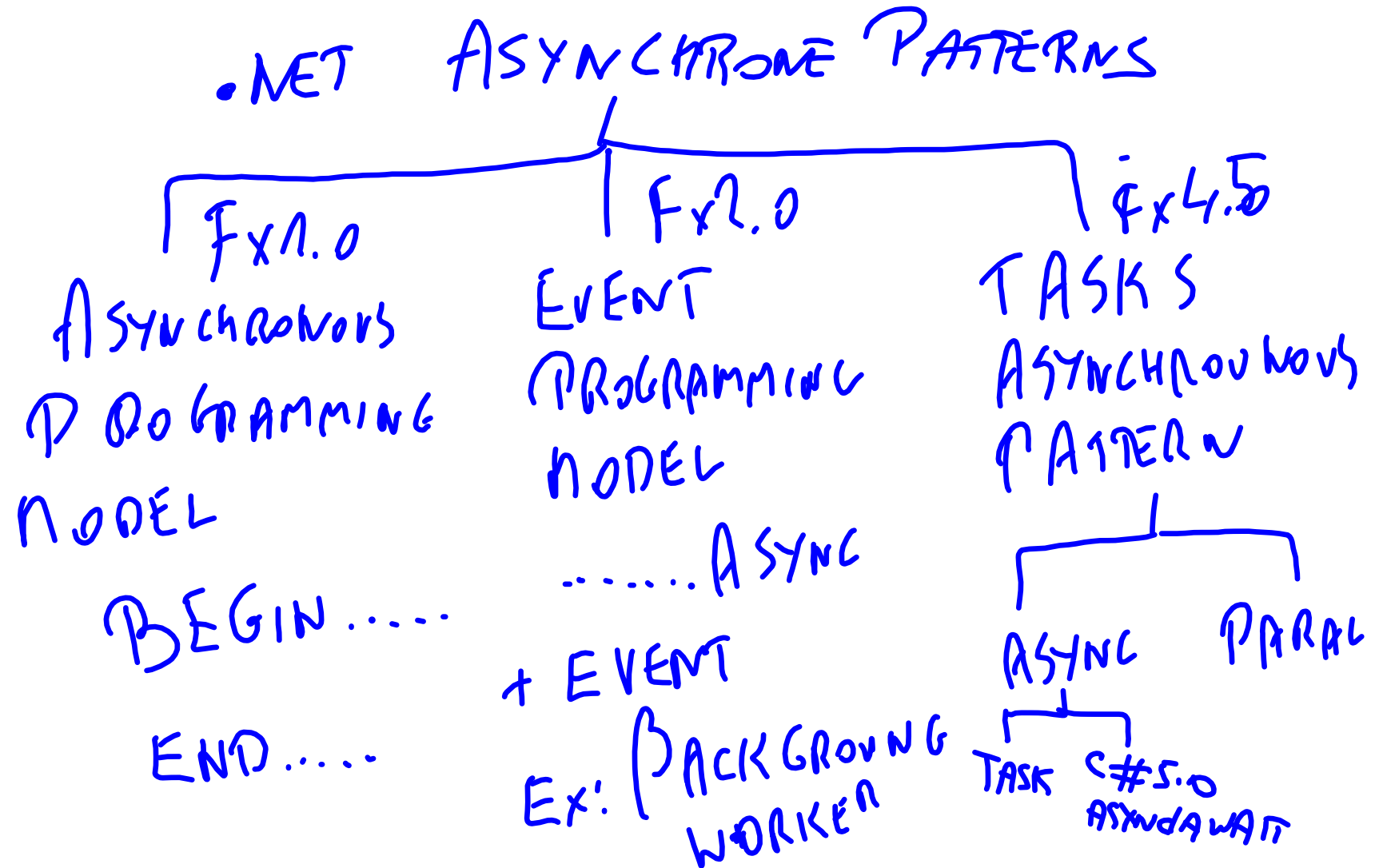
# PROCESS











- SOLID

- IoC (CONTAINER)

- REST api

- WEB API 2 + CORE

- WCF

- DESIGN PATTERN

- ARCHITECTURE LAYER

- DESIGN UNIT TEST (MOCK)

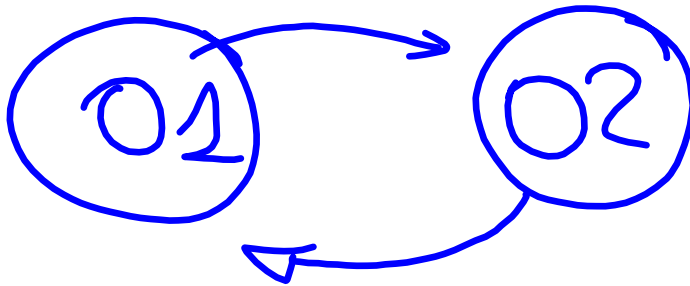
- ORN
  - CONFIRST
  - DB MODEL
- LING

TIME SHARED

T3      T2      T1

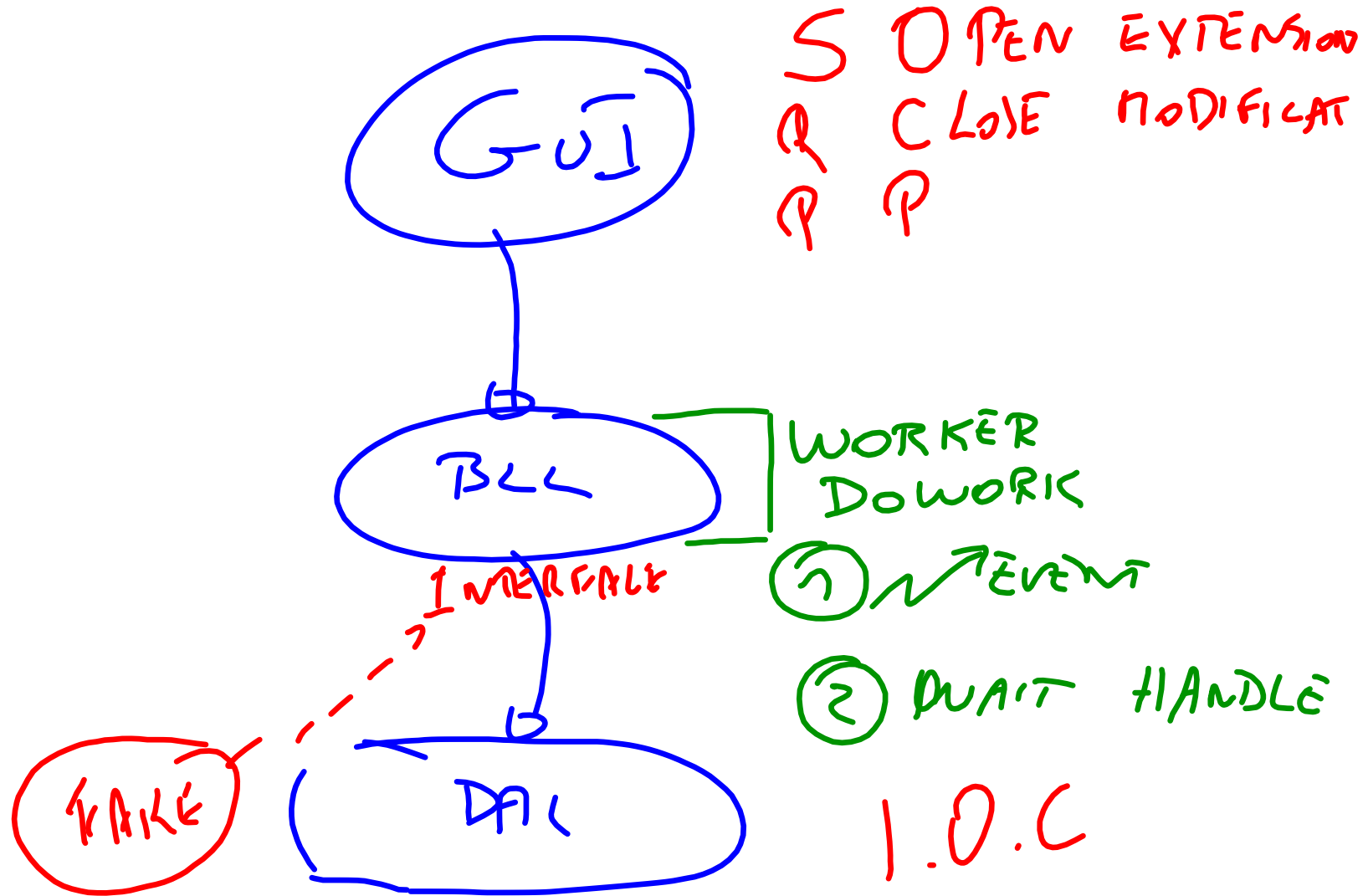
10      11      12

16      15      14





UNE HISTOIRE  
— A SYNCHRONÉ  
AFFICHE LES RÉSULTATS  
FIN DU PROGRAMME



① EVENTI necc site des data

- CREER UNE CLASSE EVENTARGS  
AVEC LES DATA

② MVC

- CRÉER UNE CLASSE VN  
AJOUTE AUX DATA DU HTML

③ ASYNC

- TRANSMETTRE DES DATA ENTRE THREAD  
↳ CRÉER UNE CLASSE POUR LES DATA.

ALGEBRE BOOLE					
$V_2$	$V_2$	$\& \cdot^X$	$  +$	$\wedge$	$-$
0	0	0	0	0	
1	0	0	1	1	
0	1	0	1	0	
1	1	1	1	1	

K I S L  
 T I M P L  
 P E P E  
 S M A T

## COMMENT LANCER UNE TASK

- ① CRÉER INSTANCE  

$$TASK\ t = \text{new TASK}() \Rightarrow \left\{ \begin{array}{l} \text{METHOD} \\ \text{TO} \\ \text{EXECUTE} \end{array} \right\}$$

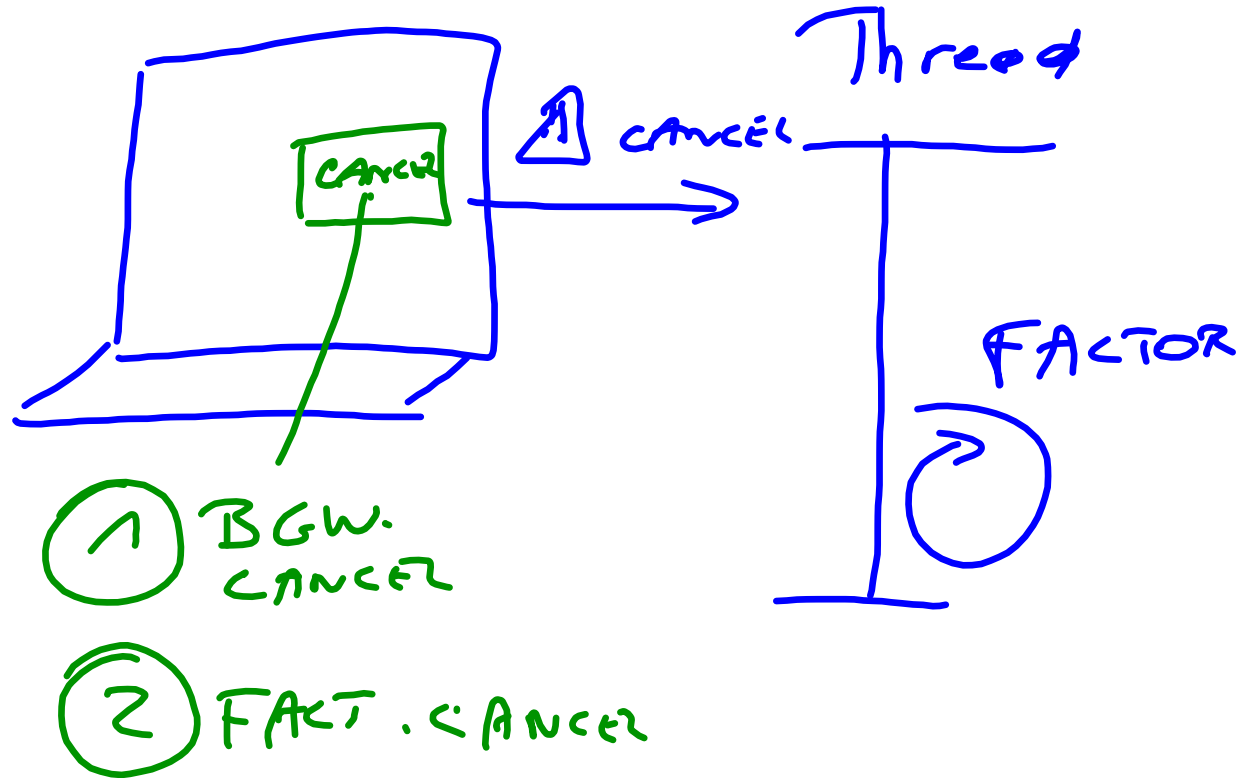
$$t.start();$$
- ② 
$$TASK\ t = TASK.run() \Rightarrow \left\{ \begin{array}{l} \text{METHOD} \\ \text{TO} \\ \text{EXECUTE} \end{array} \right\}$$
- ③ 
$$TASK\ t = TASK.FACTORY.startNew( () \Rightarrow \{ \text{MET} \} ), \text{options}$$

## COMMENT ATTENDRE LA FIN DE TASK

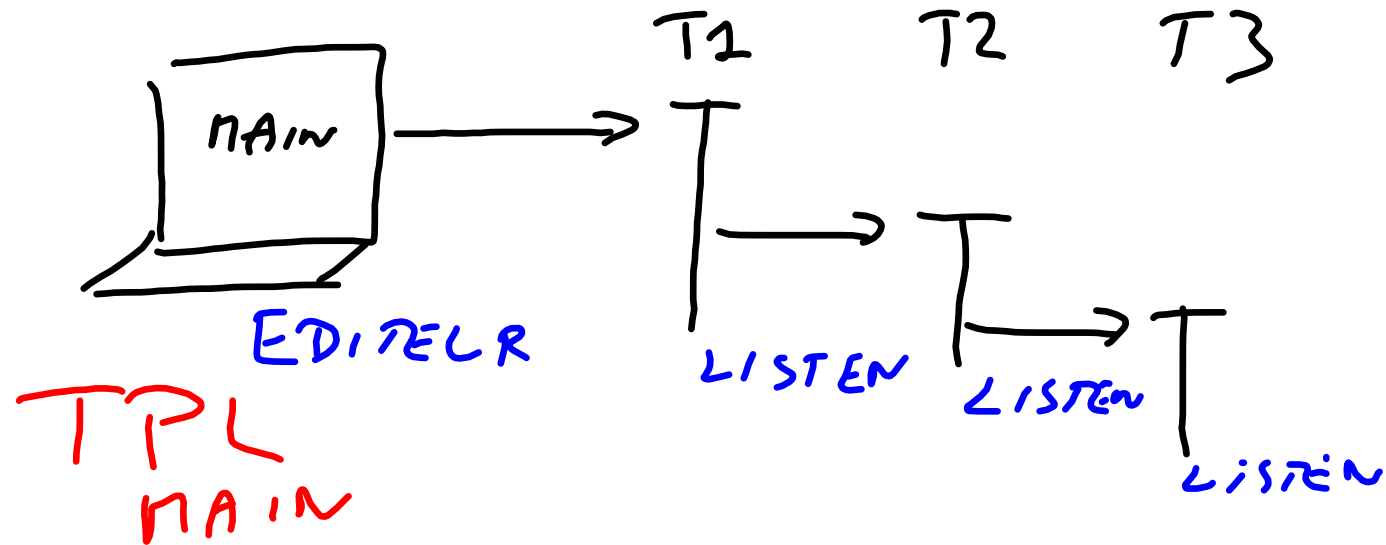
① t.wait();

② TASK.waitAll(  
                    TASK  
TABLEAU)

= TASK.waitAny(  
                    TASK  
TABLEAU)



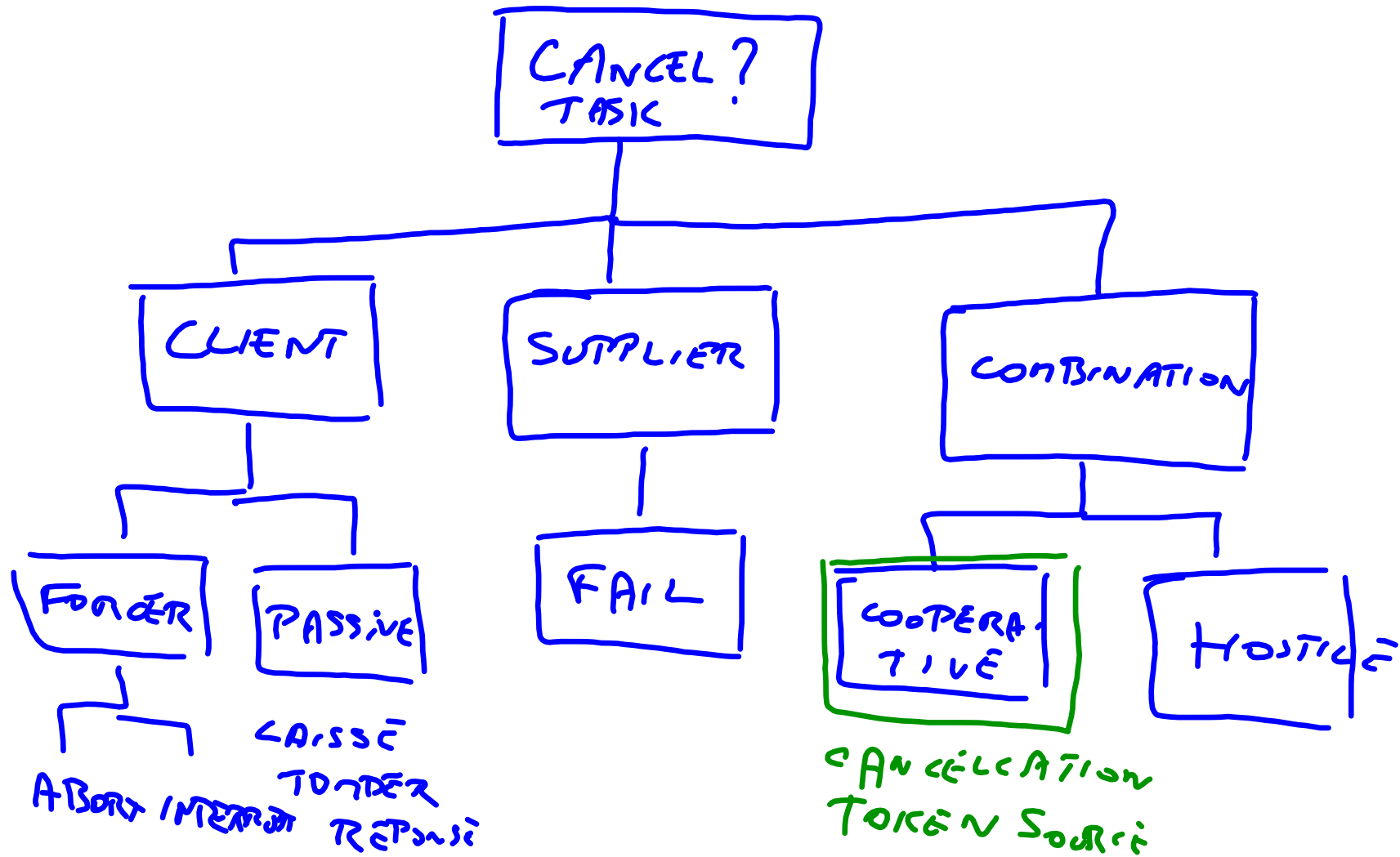




① CRÉER SOURCE

② PASSER TOKEN DE LA SOURCE

③ LANCER CANCEL



# SOL WPF ASYNC CANCELLATION

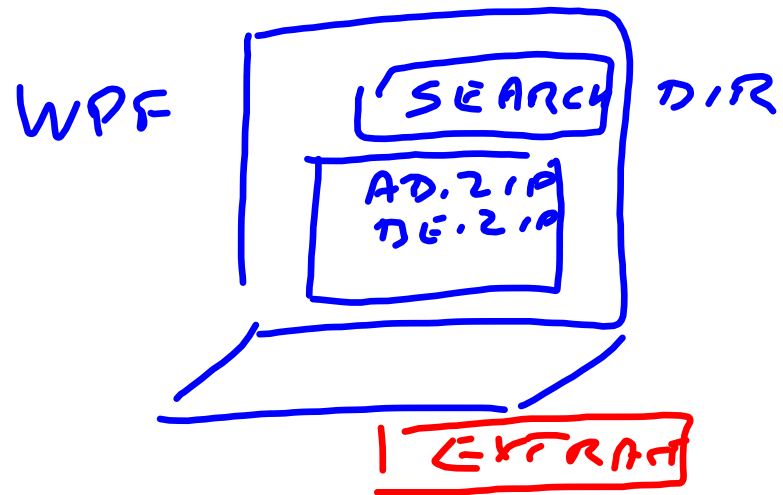
PROJECT  
COPIE WPF

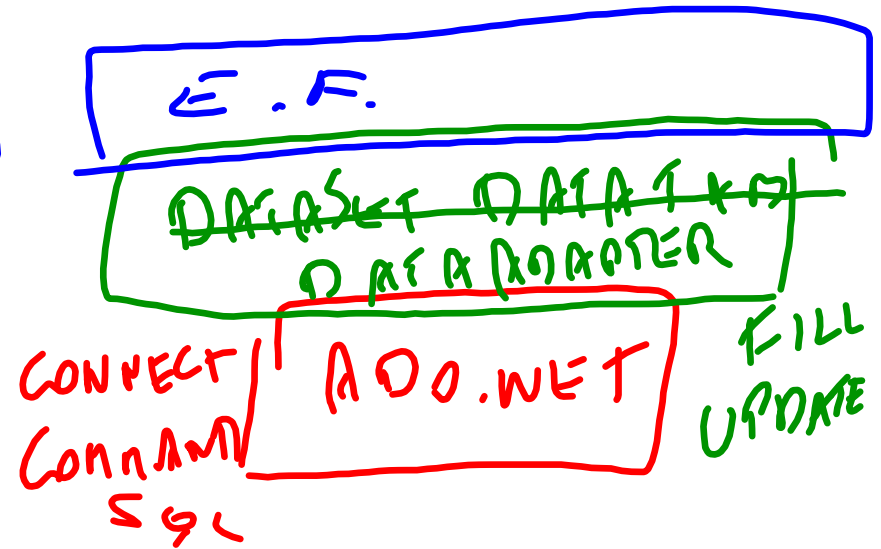
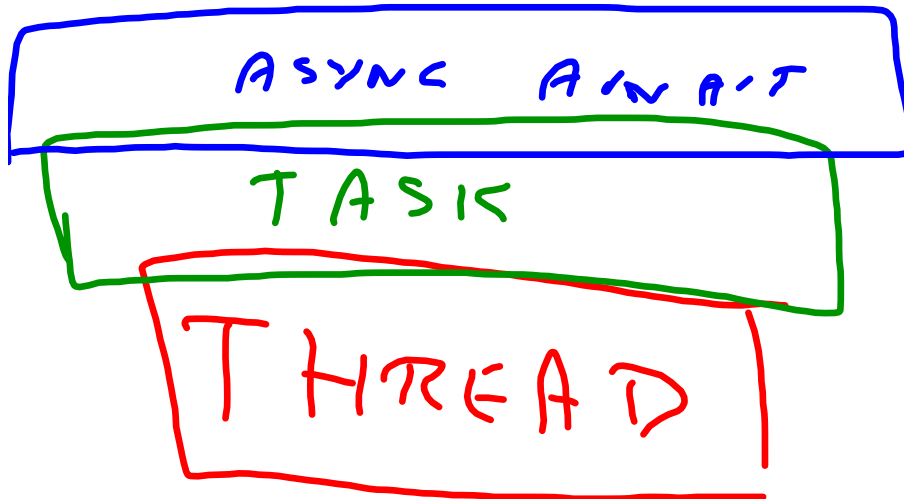


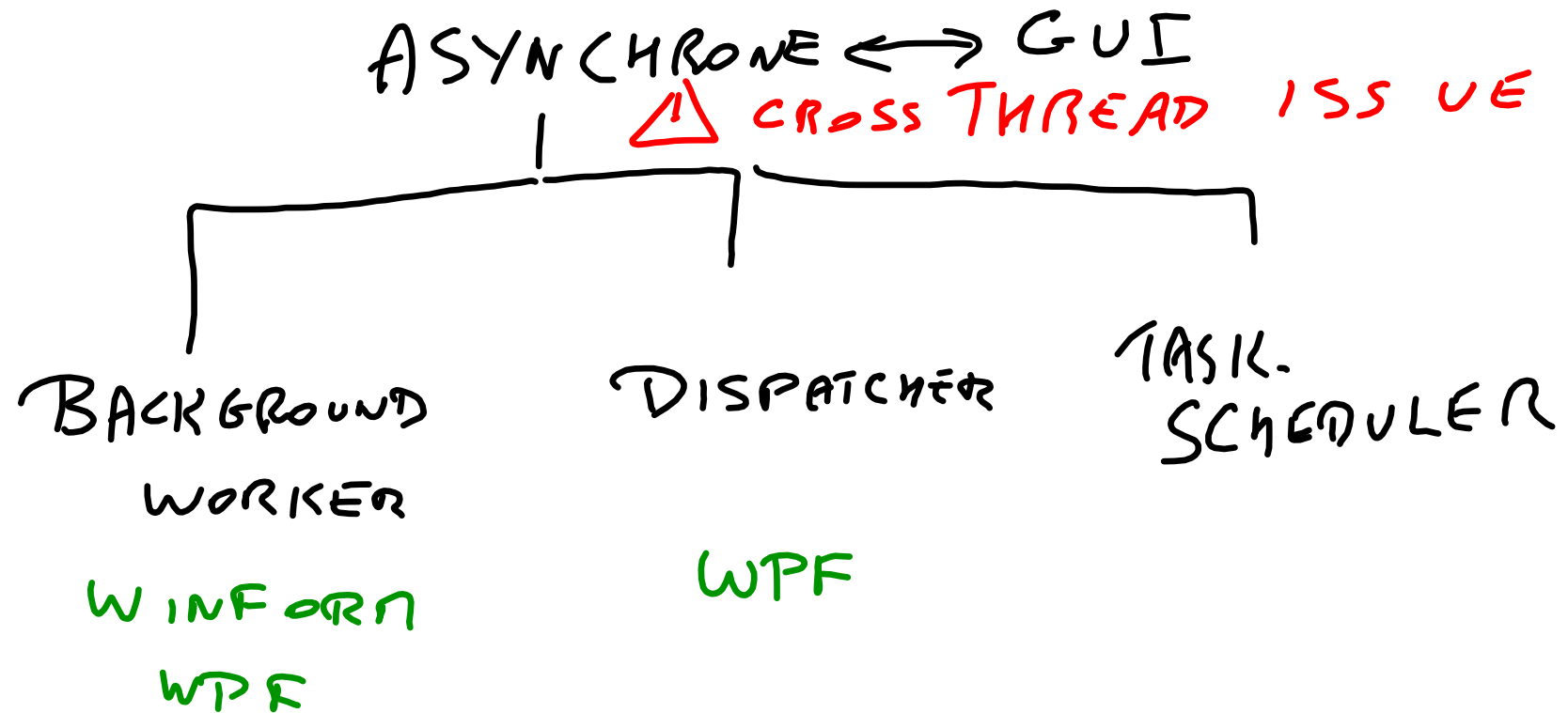
TOKEN NODE AU CHOIX

PROJECT

COPIE MATH LIBRARY







$$M1(p1, p2, p3)$$

$$M1(p1=18, p2, p3=15)$$

$$\text{CALL } M1(12, 15, 75)$$

$$\text{CALL } M1(5)$$

## C# 5.0 ASYNC / AWAIT

### LIMITATIONS:

- EN CONSOLE MAIN PAS ASYNC
- AWAIT PAS DANS CATCH NI FINALLY  
NI LOCK, NI UNSAFE CODE
- NOT REF OU OUT dans ASYNC FUNCT.
- ASYNC signifie 50 TIMES PLUS LENT