Machine coding - Splitwise

1. IoC → Spring Boot
   Dependency injection

2. Passwords
   - Plaintext
   - Encrypt
   - Encode

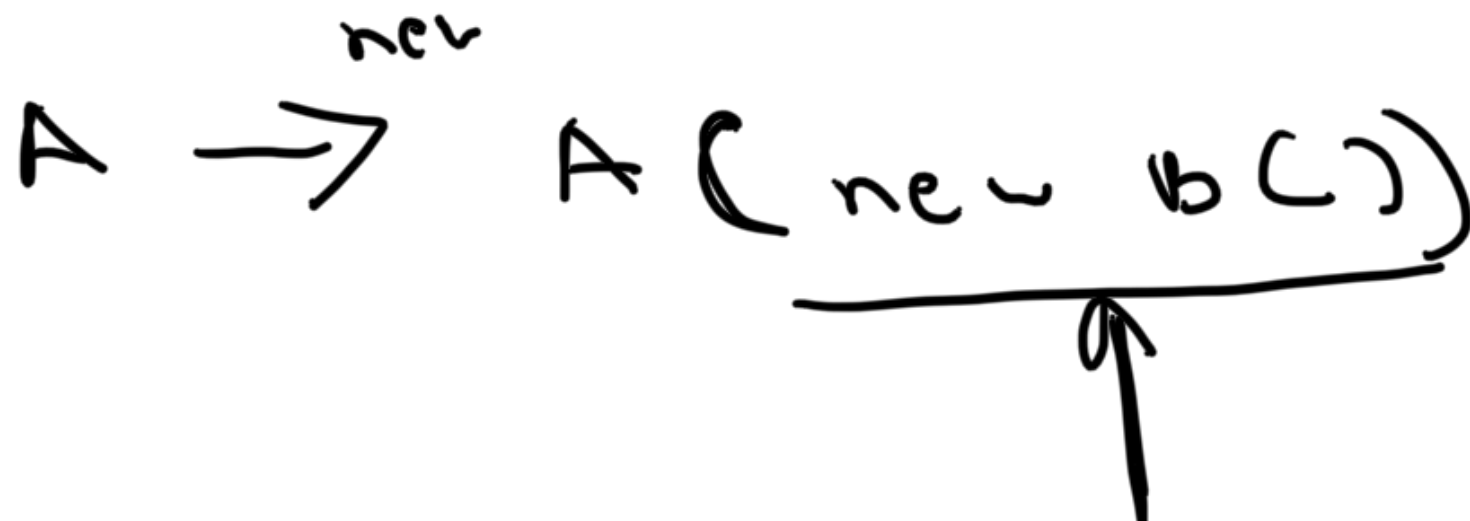3. Command Design pattern

4. S...

IoC, Dependency, Autowiring

class (A) {

    B b;

}

class B {

    C c;

}

A $\xrightarrow{\text{new}}$ A (new b())

b $\rightarrow$ new b (new C())

- Injecting dependcies on your
  Own becomes unmaintcineble

dependency $\Rightarrow$ A $\rightarrow$ b

```
class A {

    b b = new B();

}
```

① tightly coupled

② B — A needs to know deps of Ⓑ

③ [Singletons]

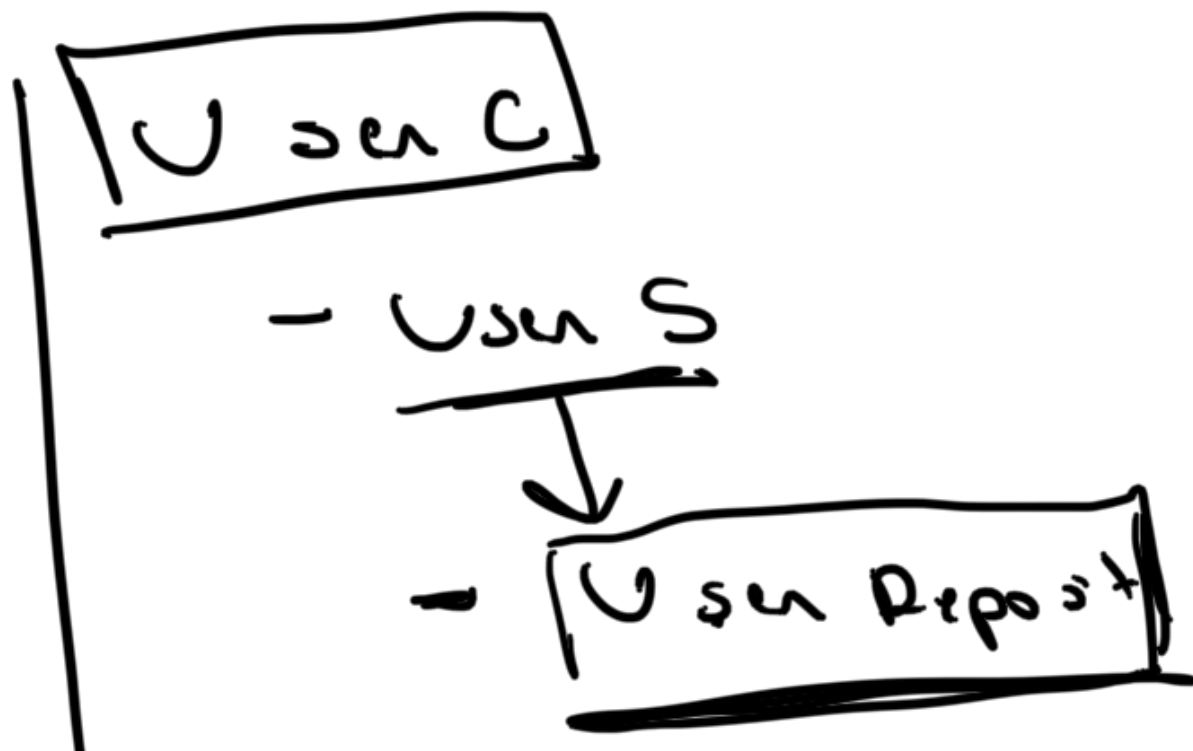# IoC + DI

Inversion of control }
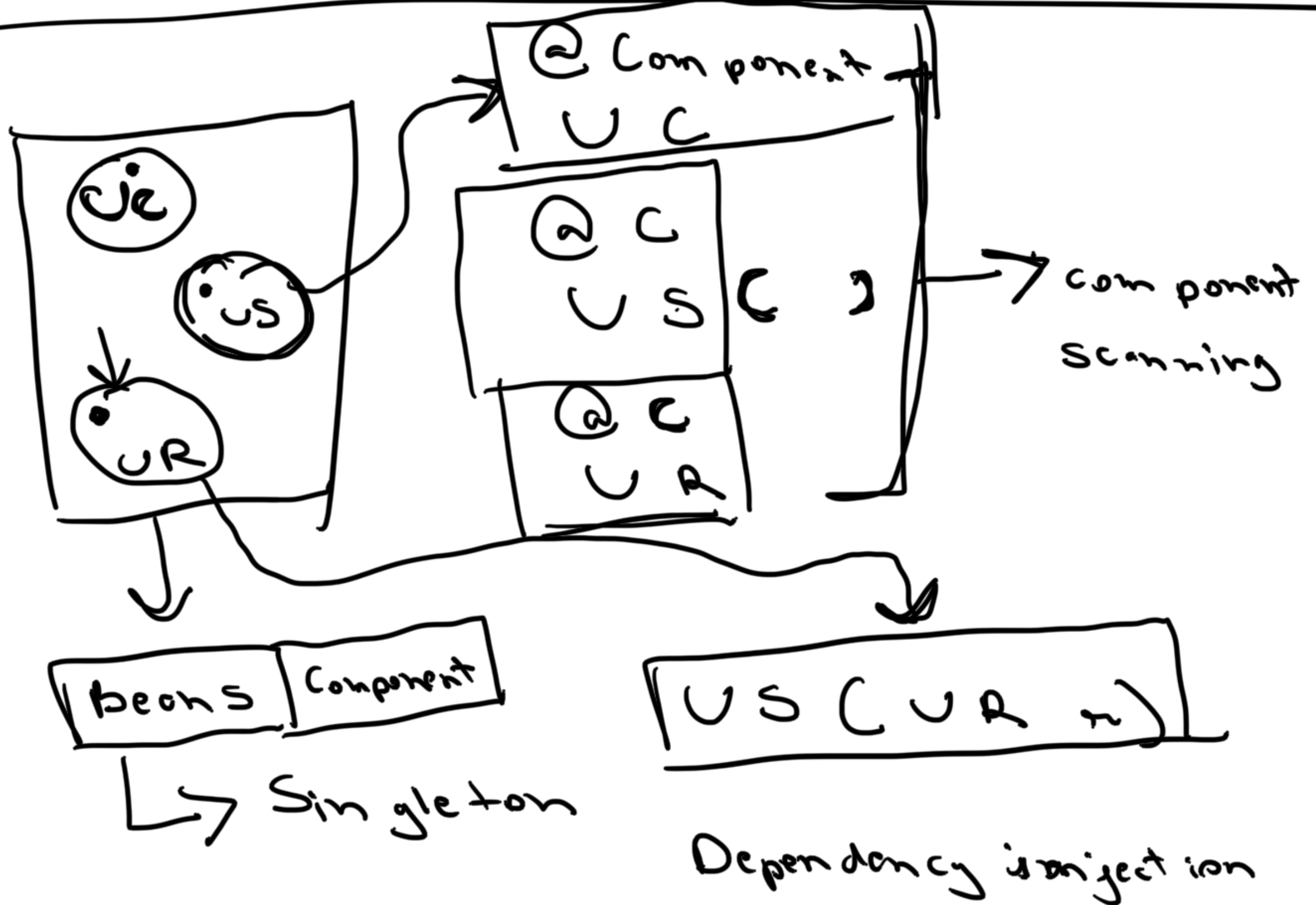Dependency injection }

IoC containers

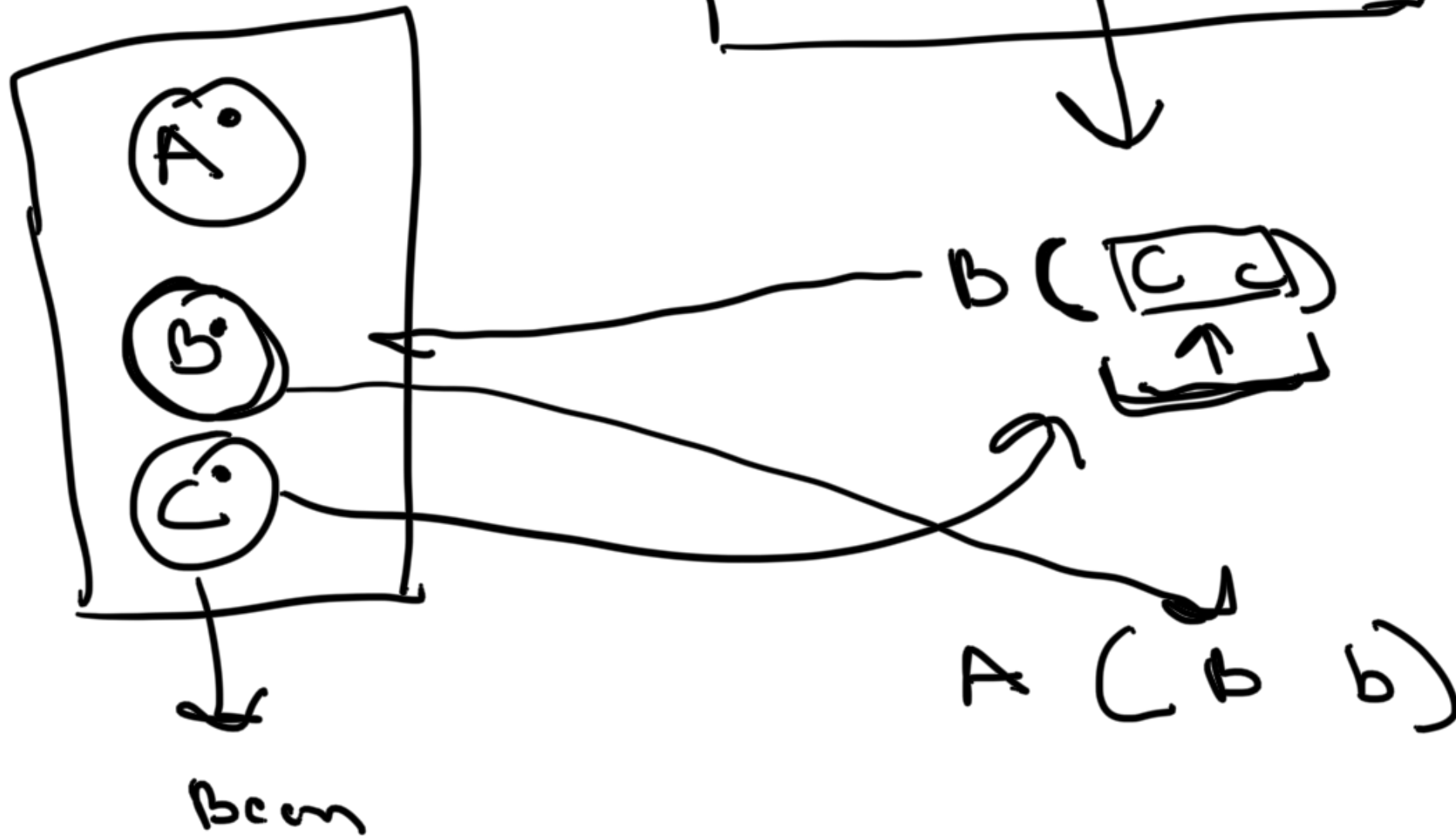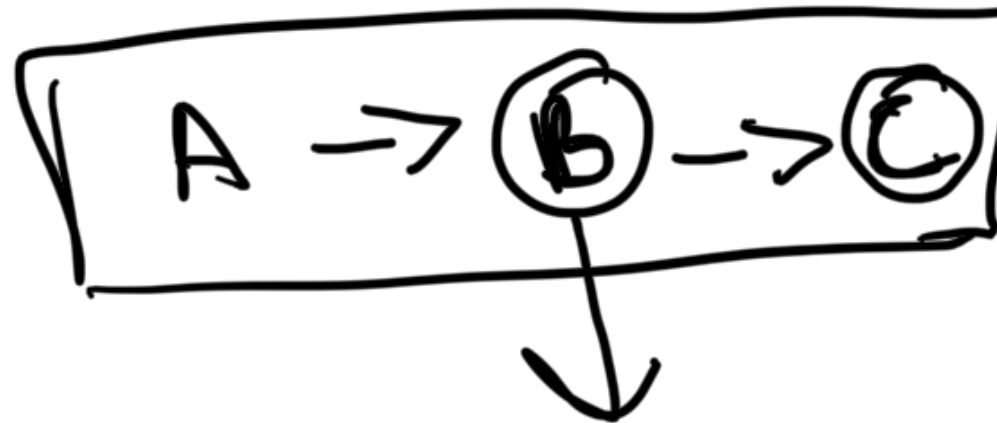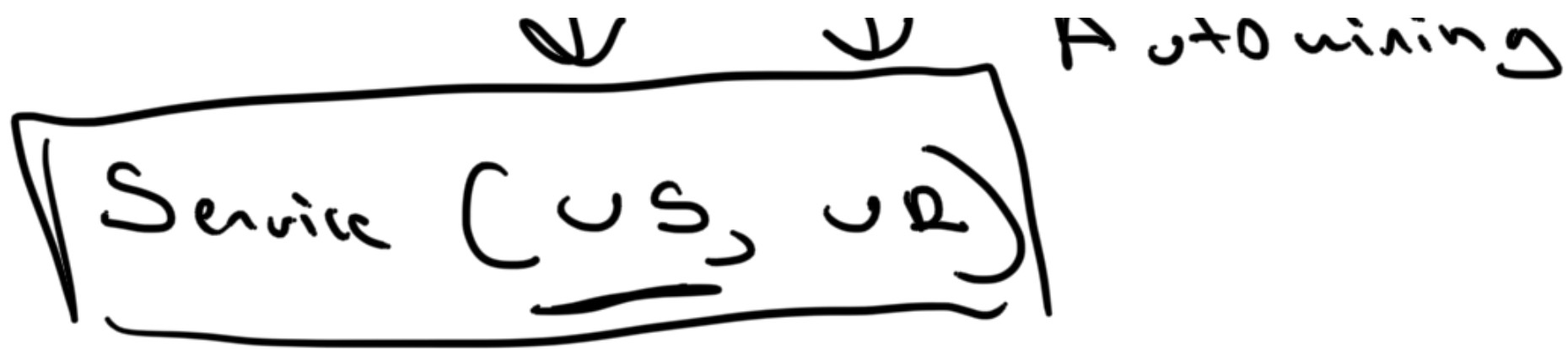A — declares its dependencies

Container — creating all instances

Container ⟶ A( ) — B

3 layers

- Controller
- Services
- Repositories

User C

- User S

↓

- User Repo

@Component
∪ C

@C
∪ S C) → Component Scanning

@C
∪ R

Beans | Component
↳ Singleton

∪ S (∪ R n)
Dependency injection

Service (US, UR) Auto wiring

A —> B —> C

A
B
C

Bean

B (C c)

A (b b)

N ((E)c)

• A   • B
• A
   • C
       • D

• E

Error
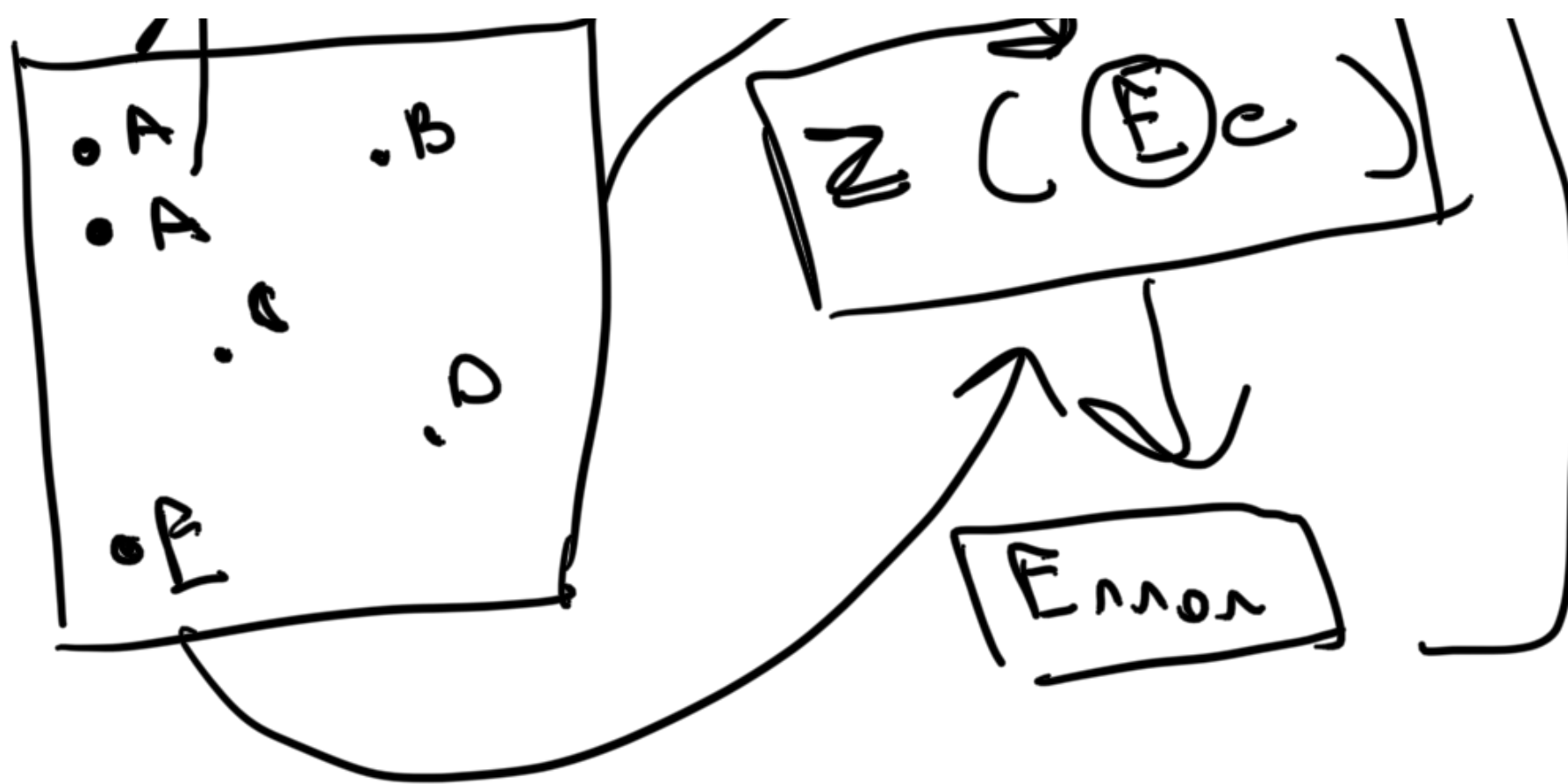
new E()
_____

→ Run time
→ Compile time
→ Build

Start Server

$\downarrow$

Component scan

$\downarrow$

Instances

$\hookrightarrow$ Validation

Controller &
Configuration

$\nearrow$ Memory

$\hookrightarrow$ Inconsistencies

| 3

→ Prototype beans

+

} → Build time

↓

+ + tle

① Component scanning

② Instance creation

③ Validation

SS

SS

SS

Port
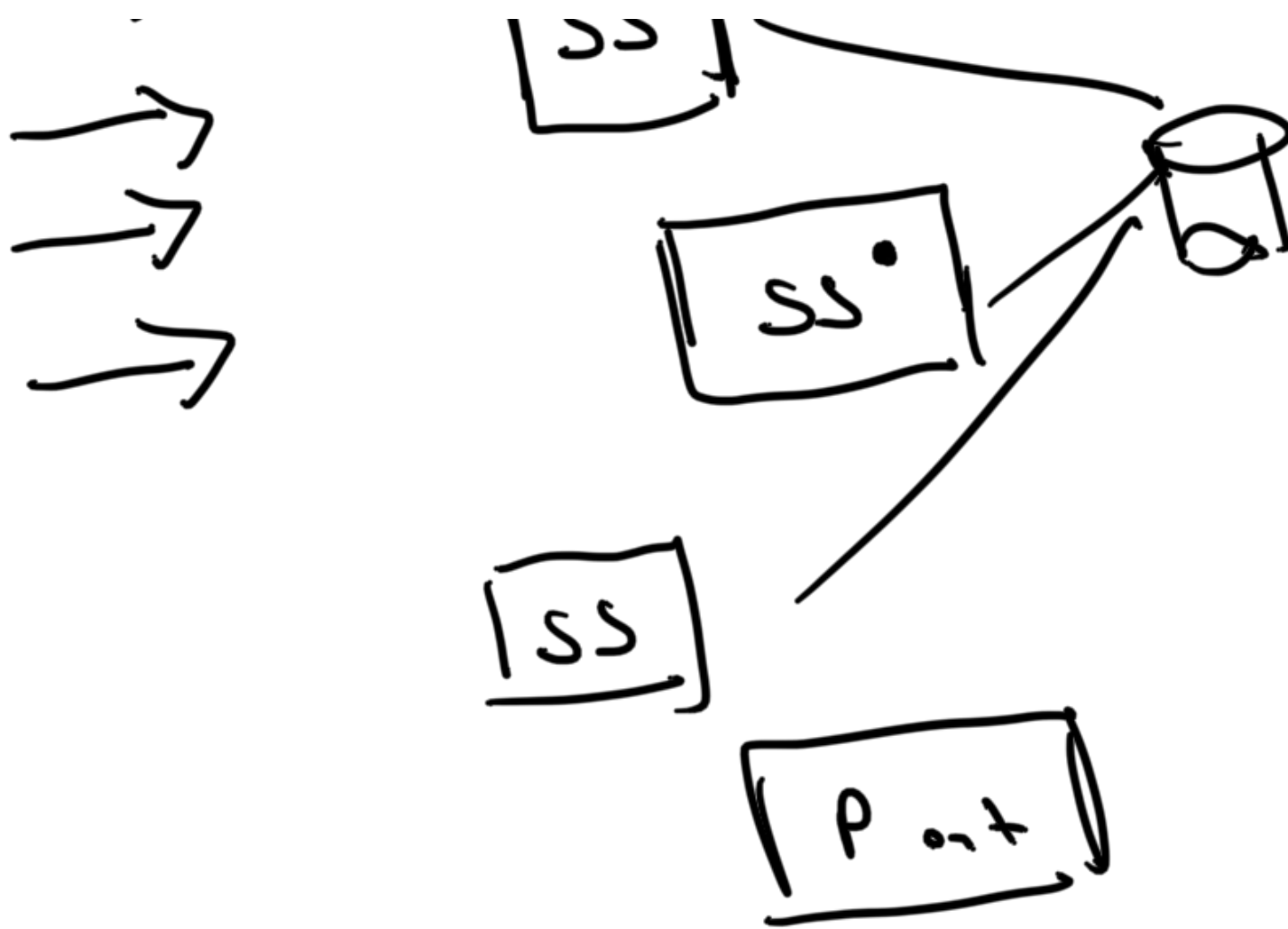
Problems

① Instance creation is

not easy

$\rightarrow$ Tightly coupled

$\rightarrow$ Singleton

② I O C

$\rightarrow$ A $\rightarrow$ b dependencies

$\rightarrow$ b $\rightarrow$ A

A {

b

}

$\nearrow$ b ( ) = instance

A (b b)

$\rightarrow$ Aspect oriented prog.

Injection

$\rightarrow$ Setter injection $\rightarrow$ @ Autowired

$\rightarrow$ field injection $\rightarrow$

$\rightarrow$ | c'tor injection |

IoC $\rightarrow$

DI

A $\rightarrow$ b

A $\rightarrow$ A
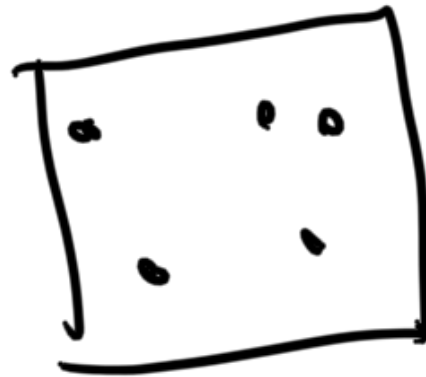
new A(new B( ))

Passwords

① Plaintext
    └→ Data breach
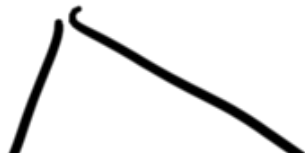
En cryption

En coding

Password → 1 2 3

* * * * * * *

password

Run length encoding

| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 3 3 |

pixel 0 - 255

| 51 | 32 | 33 |

A → ⚡

B → ②

Public

Password

ALG → ✳✳ ✳✳✳

RSA 256 | 512

Private

✳✳✳✳ ✳✳
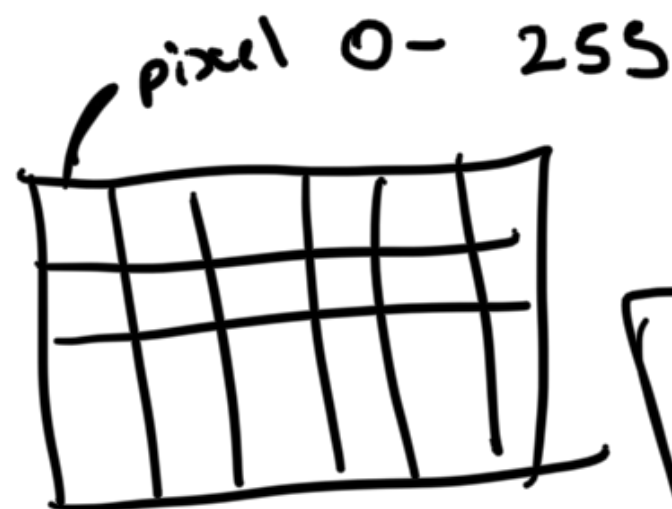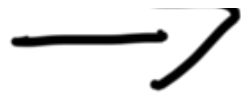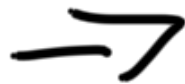
ALG → Password

Passwords

$\rightarrow$ private key

public

private

C ——— S ⟳

① User $\rightarrow$ Client $\rightarrow$ pw

② public + pw ⟶ [encrypted]

③ client ⟶ server ⟶

④ Db ⟶ server ⟶ enc. pw

⑤ private + en. pw

⟶ plein pw

browser

"secure" + public

⟹ "****" — Db

"***" + [Private] — Server

=

"Secure"

{ DB
  File } ⇒ Secrets Manager
         ⇒ File Vault

User

Private Key

[***] [1254] →
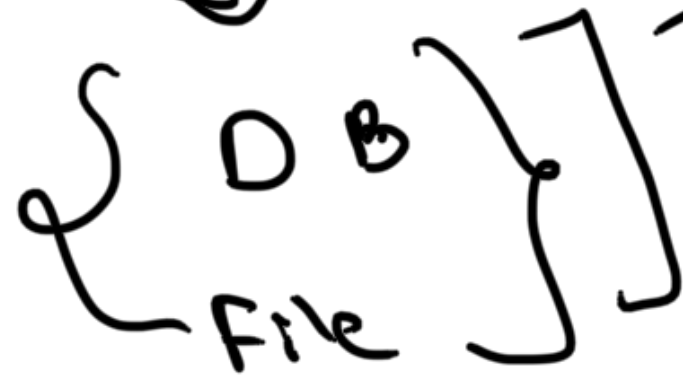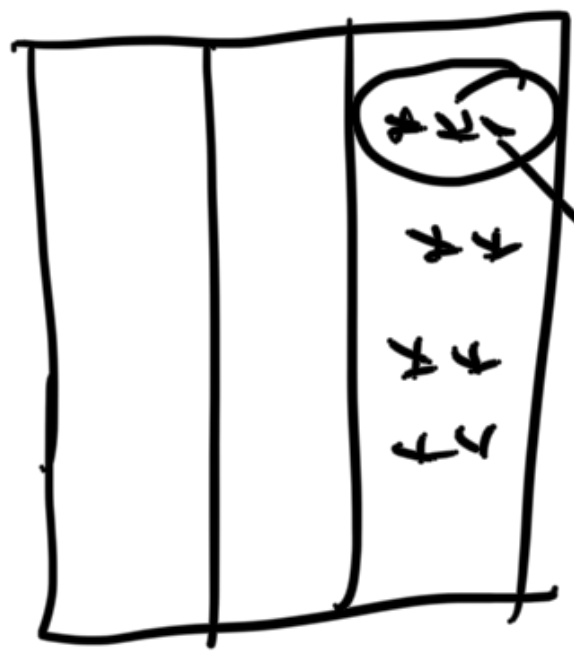
plain text

② Slow

③ Extensibility not great

Encoding

$pw \longrightarrow$ [∽p]

$pw \longrightarrow nd S(pw)$

Salting

93195	Pw

→

***** *

algo
Salt

Db
Sm

→ Salt

→ pepper }

	Pw

encoded   Encoded
Pw

(Sel)
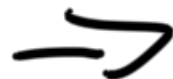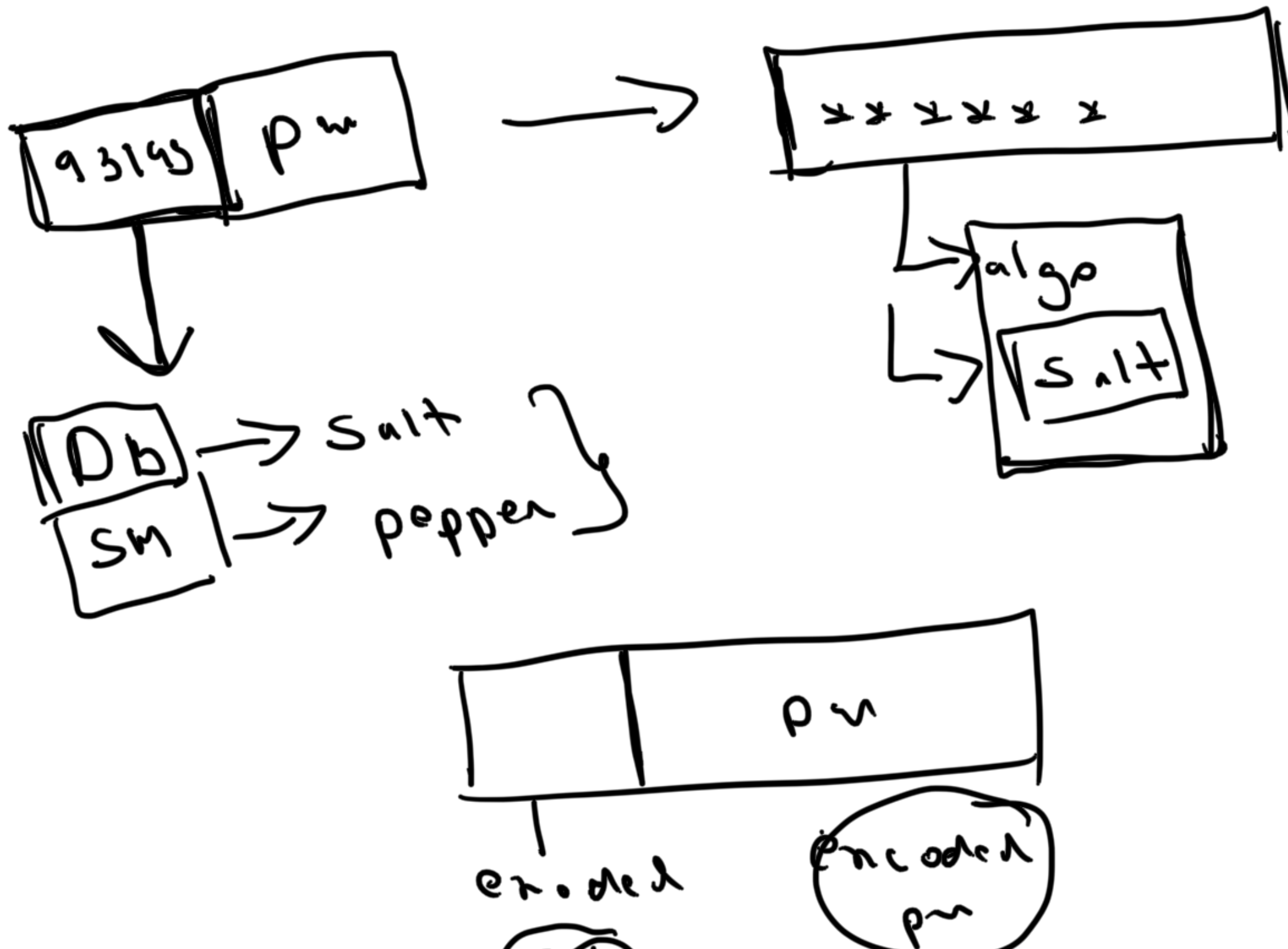
| 123 pass | ——— encode('123pass")

Solt ——>) pw

1 put beg ——> | All pw |

⏝ Sen

Enco

Encod.

BCrypt