zeptolab

Viktor Korsun

input:

Many studios are implementing their containers.
Many libraries contain their own implementations.

What STL is:

Alex Stepanov, interview to Graziano La Russo.

What STL is:

STL is not object oriented.

I think that object orientedness is almost as much of a hoax as Artificial Intelligence. I have yet to see an interesting piece of code that comes from these OO people.

What STL is:

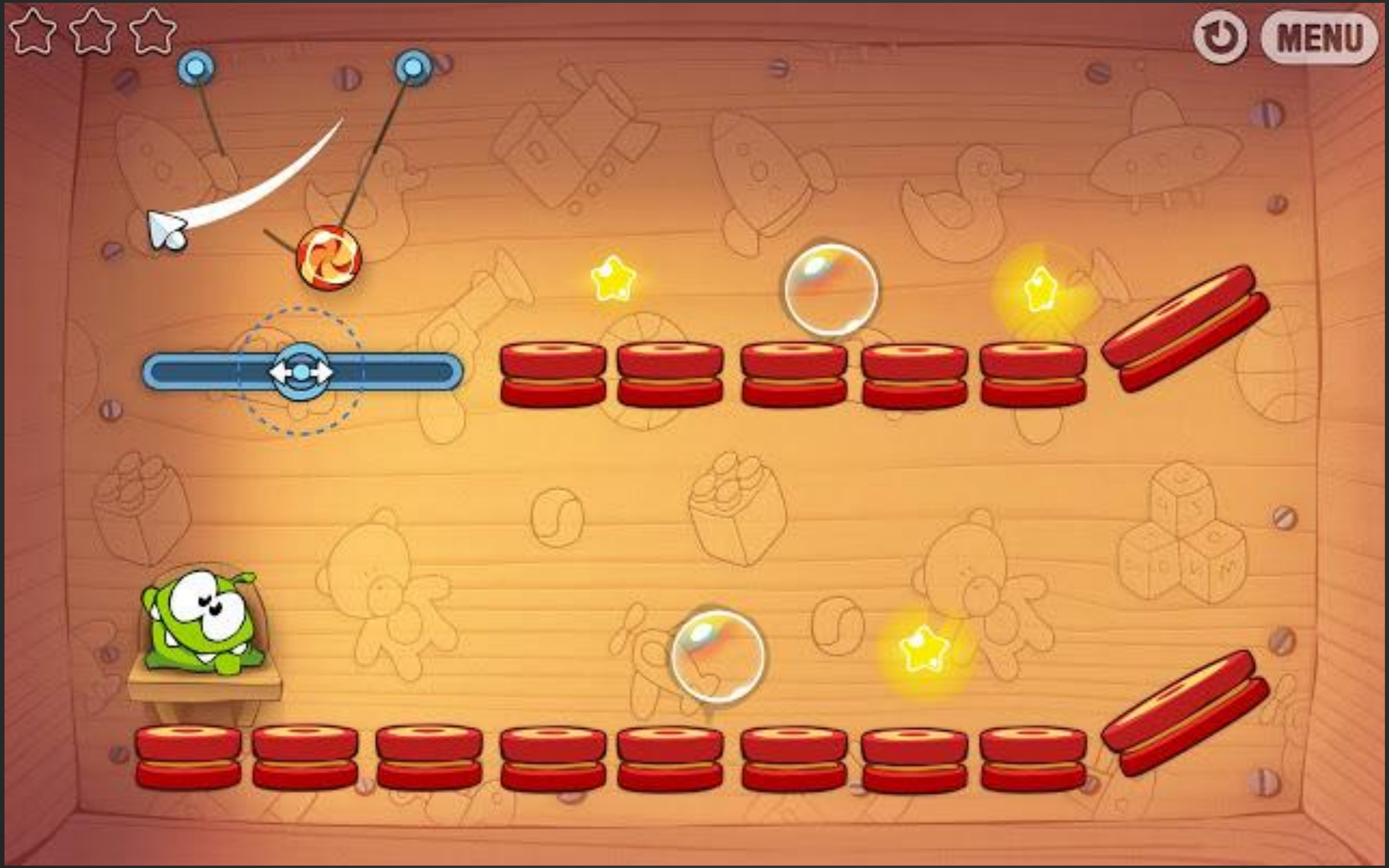
I find OOP technically unsound. It attempts to decompose the world in terms of interfaces that vary on a single type.
To deal with the real problems you need multisorted algebras – families of interfaces that span multiple types.

I find OOP philosophically unsound.
It claims that everything is an object.

Even if it is true it is not very interesting - saying that everything is an object is saying nothing at all.

I find OOP methodologically wrong.

Object-orientedness



OOP Principles

- Abstraction
- Incapsulation
- Inheritance
- Polymorphism

What we need

- Insert an object
- Address an object
- Remove an object

- Ideally:
  - Have polymorphism out of the box
  - Support polymorphism of the container

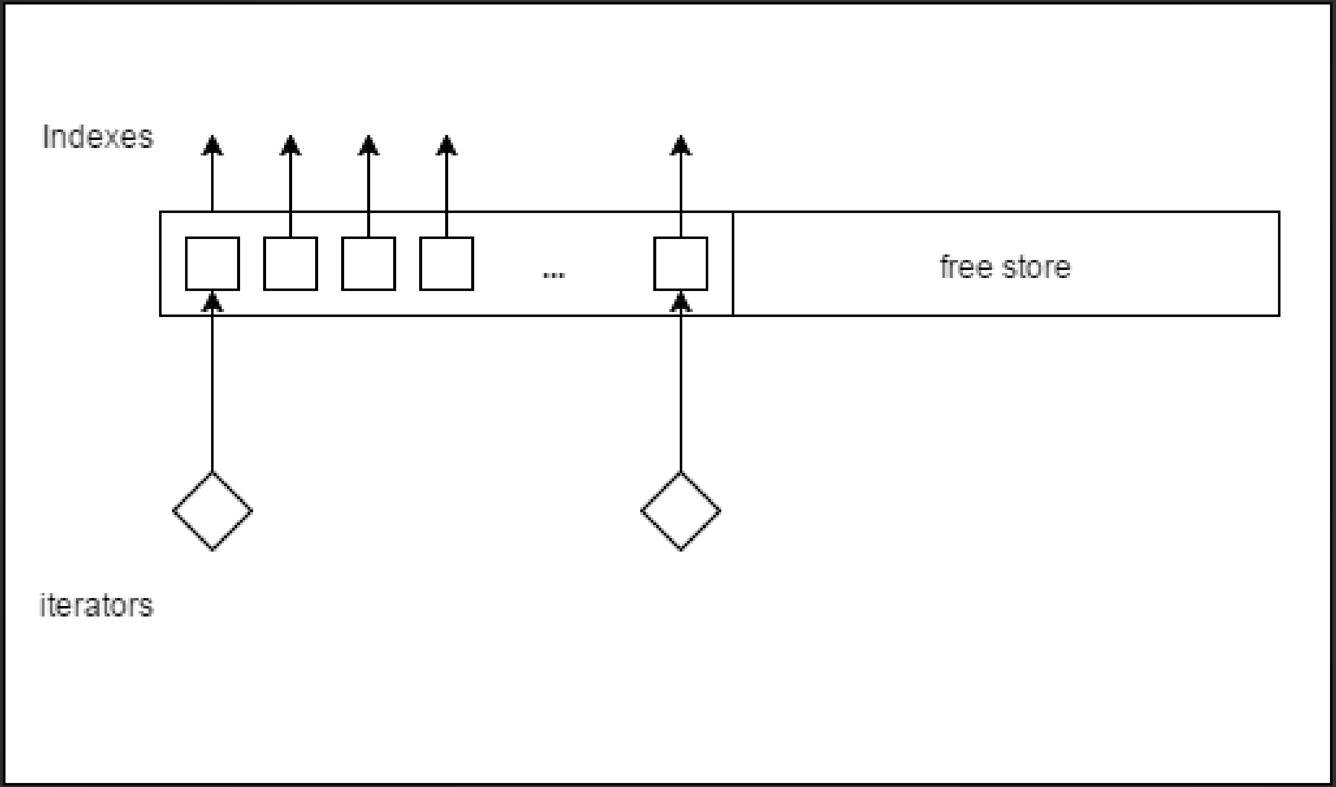
Generic synthax

```
class A
{
//...
};

class B : public A
{
//...
};

ovector<A> vec;
vec.push_back(A());
vec.push_vack(B()));
```

What we do not need

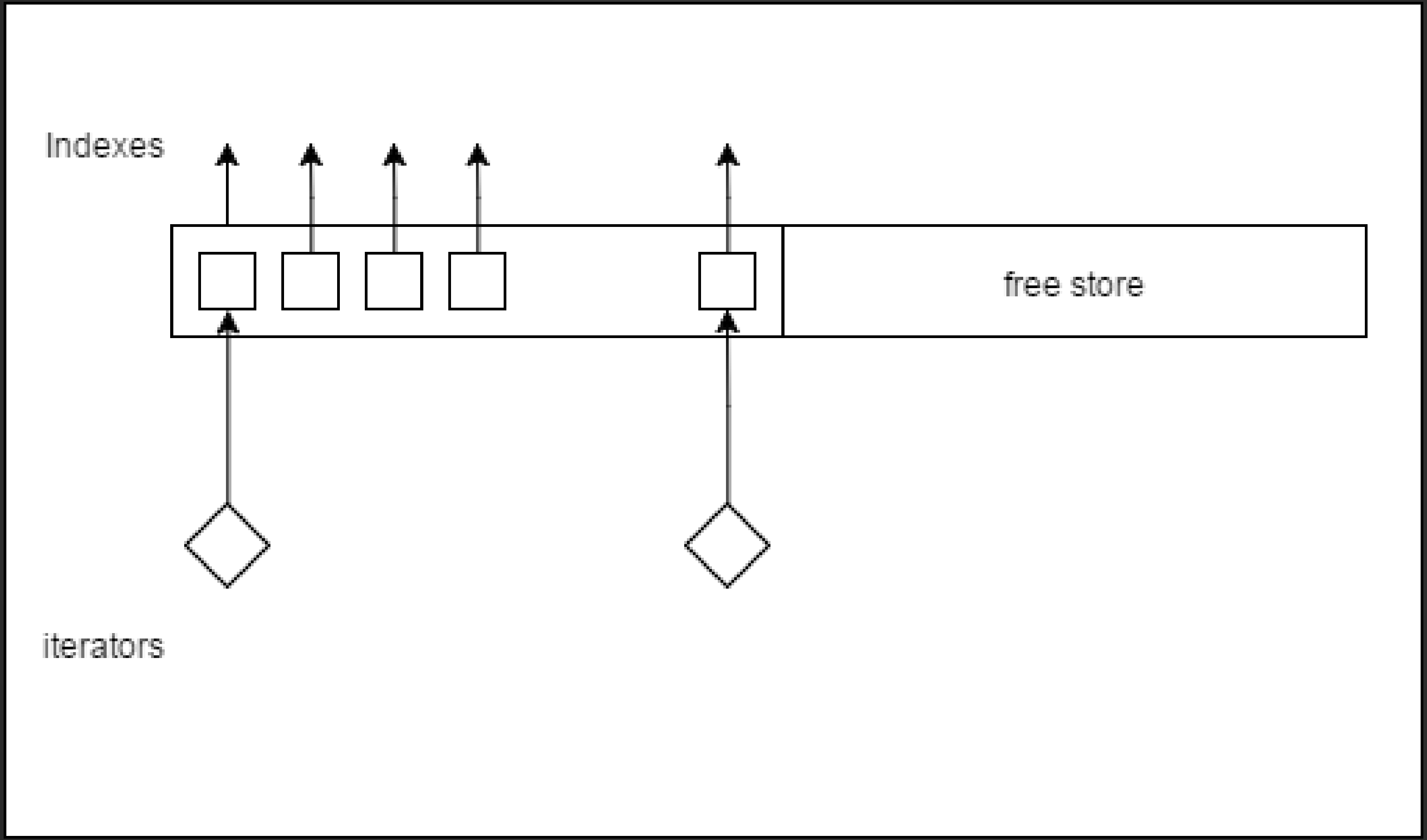
Rely on objects memory layout
(we have abstractions)

Usage of Vector

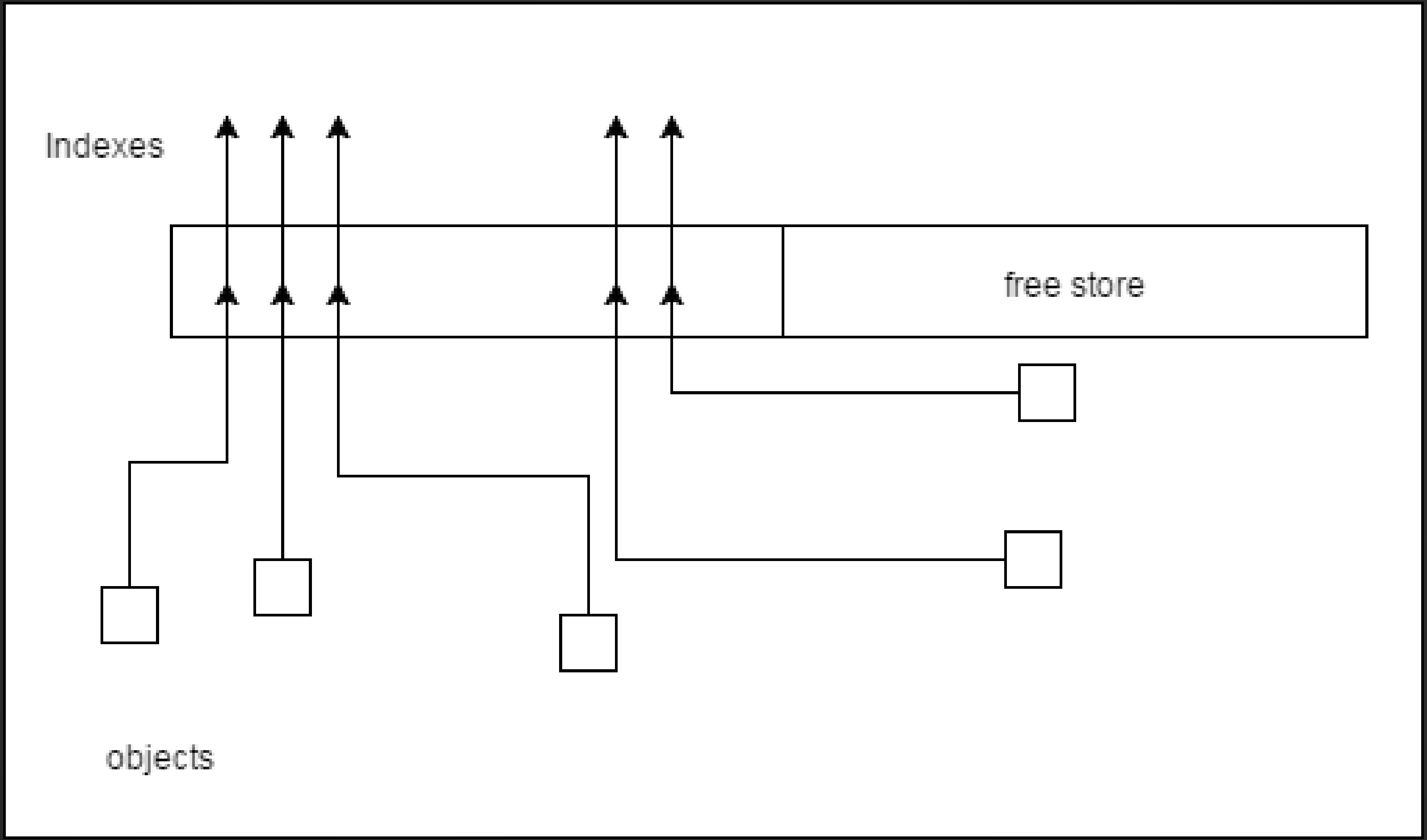


Usage of Vector

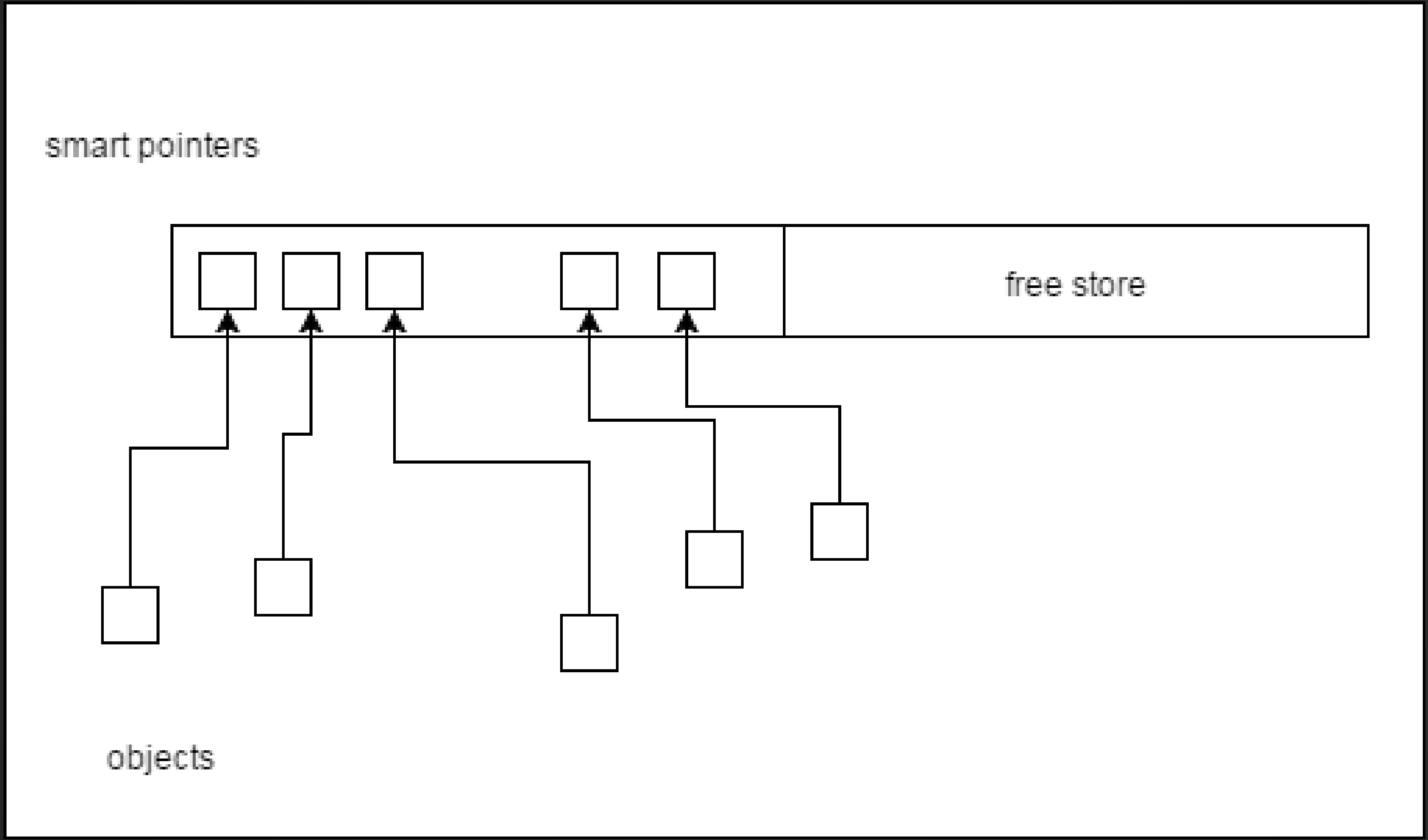
Usage of Vector



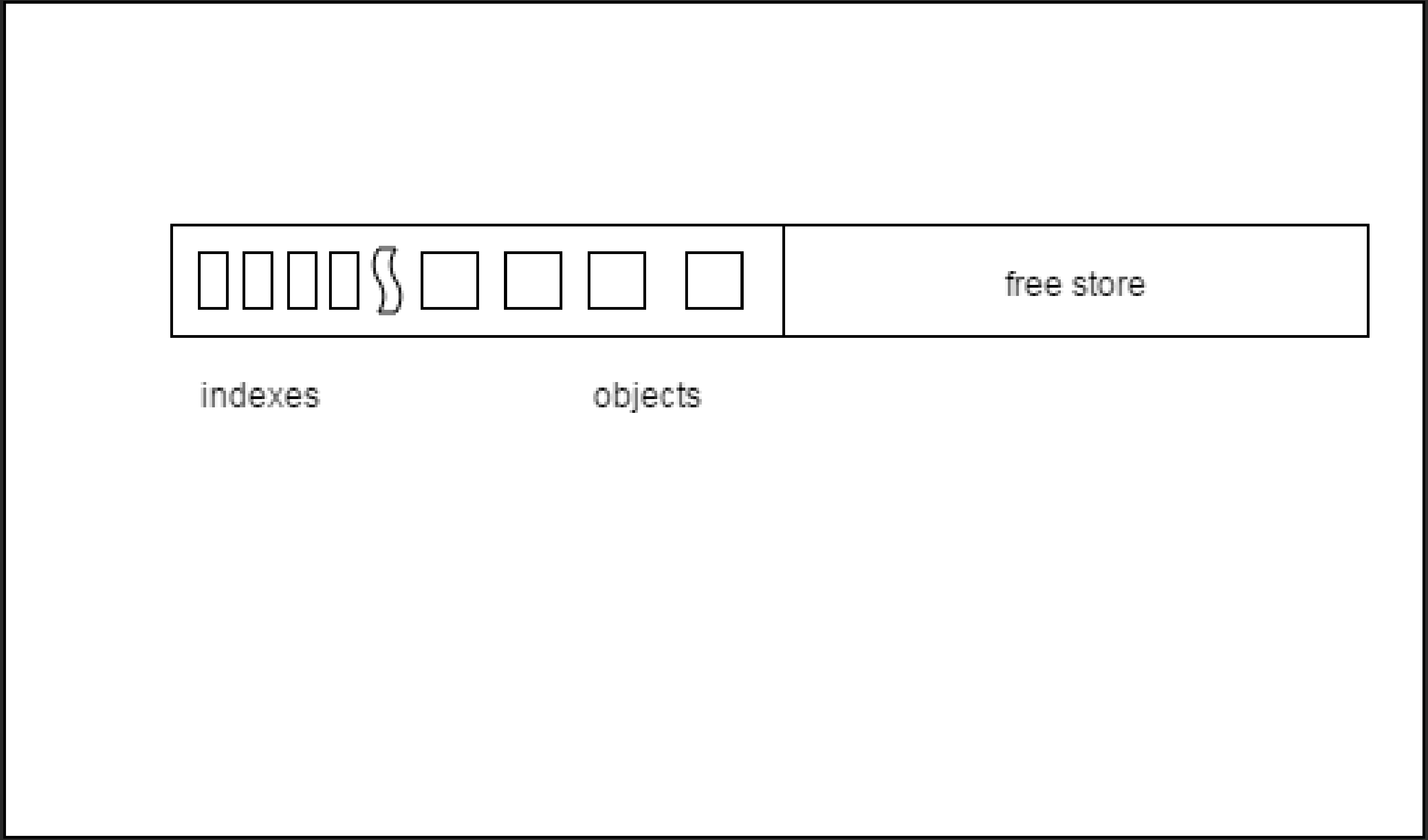
Vector of pointers



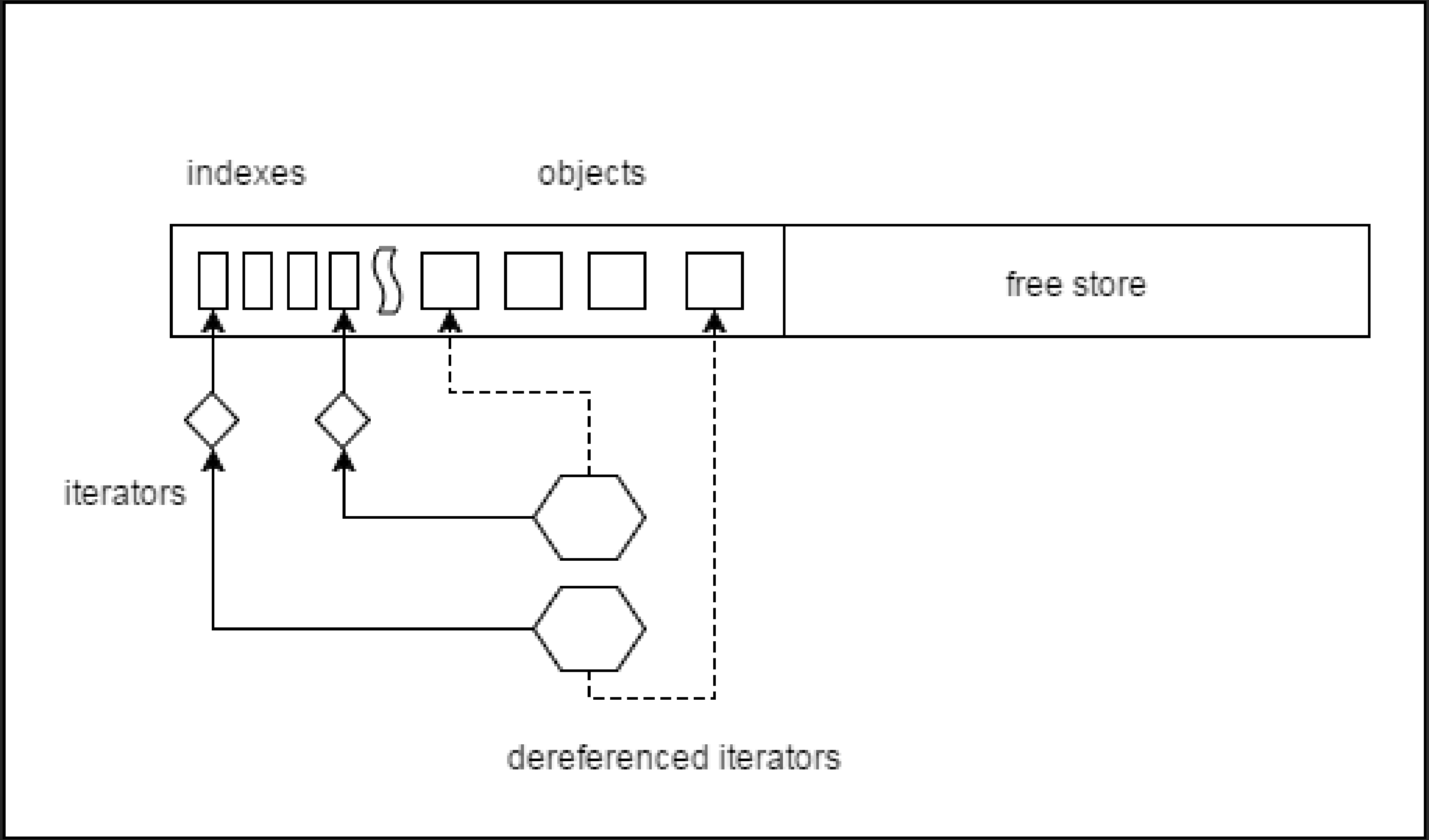
Vector of smart pointers



Suggested model



Working with <algorithm>



Efficiency

demo

Class hierarchy

Q: A frequent dilemma for me was:
should I design this function as a member function or as a generic (global) function?
What has been the rationale of this decision in STL?

A: Make it global if it at all possible.
It would be much nicer if begin and end were global - it would allow us to define them for C arrays.
It would be so much nicer if operator* was global with the default definitions.

Generic synthax

```
std::set<std::string> game_objects = gamescene.getgameobjects();
…
std::find(game_objects.begin(), game_objects.end(), std::string("omnom"));
```

Generic synthax

```
std::set<std::string> game_objects = gamescene.getgameobjects();
…
game_objects.find(std::string("omnom"));
```

Final quiz

How to make the code faster?

Final quiz

```
constexpr int N = 1e3;

char arr[N];

for (int i = 0; i < N; i++)
    arr[i].c = fibonacci(i) % 32;
```

viktork@zeptolab.com