

# 有前端思想的物联网架构

@aimingoo

aimingoo@nanchao.org





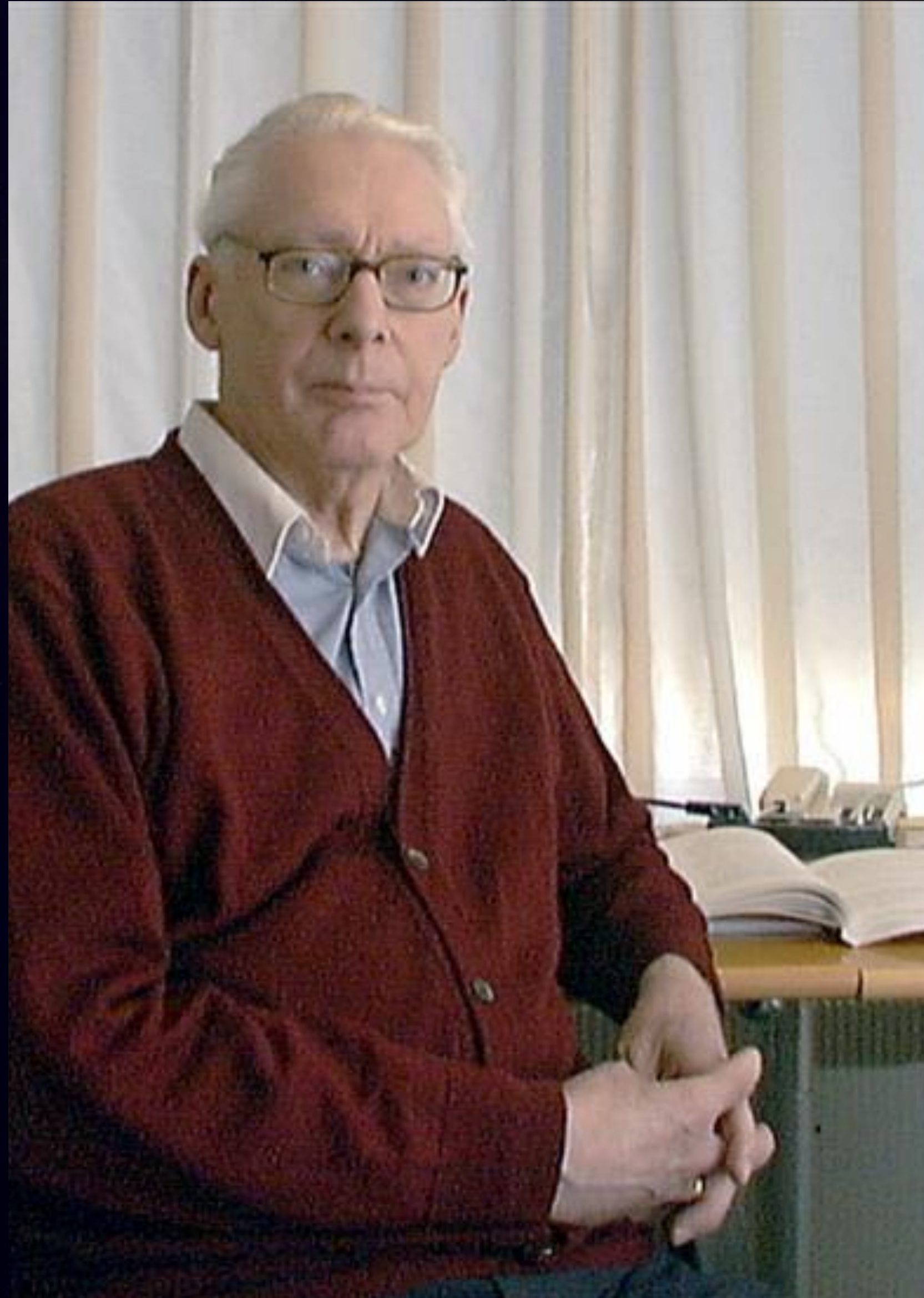
Edsger Wybe Dijkstra  
1930.05.11~2002.08.06  
@1972

为我们理解程序语言的结构，表示方法与实现做出了巨大的贡献。



Tony Hoare  
1934.01.11~  
@1980

程序语言定义与设计领域的根本性的贡献。



Ole-Johan Dahl  
1931.10.12~2002.01.29  
@2001

在设计编程语言SIMULA1和SIMULA67时产生的基础性想法，这些想法是面向对象技术的肇始。

《结构程序设计札记》

《数据结构札记》

《层次程序结构》

} 程序 = ?



# 程序是可被组织的元素

(程序的可结构化特性)



<HTML>

...

<BODY>

...

</BODY>

</HTML>

*Handwritten signature*

A vibrant nebula with red, orange, and yellow filaments against a dark starry background. The nebula's structure is complex, with wispy, interconnected strands of gas and dust. The colors transition from deep red on the right to bright yellow and orange in the center and left. Numerous small, multi-colored stars are scattered throughout the field of view.

`<IMG src="" . . .`

A small, stylized signature or logo in the bottom right corner, rendered in a light, reddish-orange color that blends with the background. It appears to be a cursive or calligraphic mark, possibly a name or brand identifier.

*[Signature]*



```
<IMG id="me" src="" class="" ...
```

```
<SCRIPT>
```

```
me.onclick = ...
```

```
...
```



What's IoT ?

<PRODUCT>

<APPLICATION service="publish-as-gw...">

<NETWORK gateway="connect-to-net...">

<DEVICE>

...

- NETWORK include DEVICES and/or sub-NETWORKs
- DEVICE include sub-DEVICES or nothing
- APPLICATION include NETWORKs
- PRODUCT is reserved

# 最小的网络设备

```
<NETWORK><DEVICE id="me" /></NETWORK>
```



me { 是什么?  
在哪儿?  
做什么?

*Handwritten signature*

```
// JSON Style, the facade of network
{
  "devices": [{
    "id": "me",
    "className": "Button"
  }]
}
```

className支持对同一个Thing层叠修饰，  
如同一个对象的多个Interface。  
因此任何设备可以『表现得』像是一个网络。

```
<NETWORK url="http://...">
```

```
...
```

```
<NETWORK name="office">
```

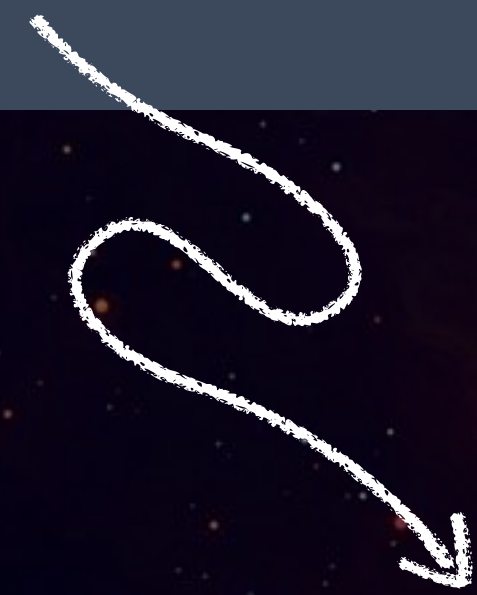
```
<NETWORK addr="上海某地的海上">
```

```
<NETWORK ip="192.168.0.121">
```

```
<DEVICE id="me" class="button"/>
```

```
<BODY>
```

```
<IMG id="me" src="" class="" ...
```



```
...
```

```
<NETWORK ...>
```

```
<DEVICE id="me" class="button"/>
```



```
...  
<NETWORK ...>  
  <DEVICE id="me" class="button"/>
```

```
{ $("device")[x]  
  $(".button" ...  
  $("#me" ...  
  ...
```



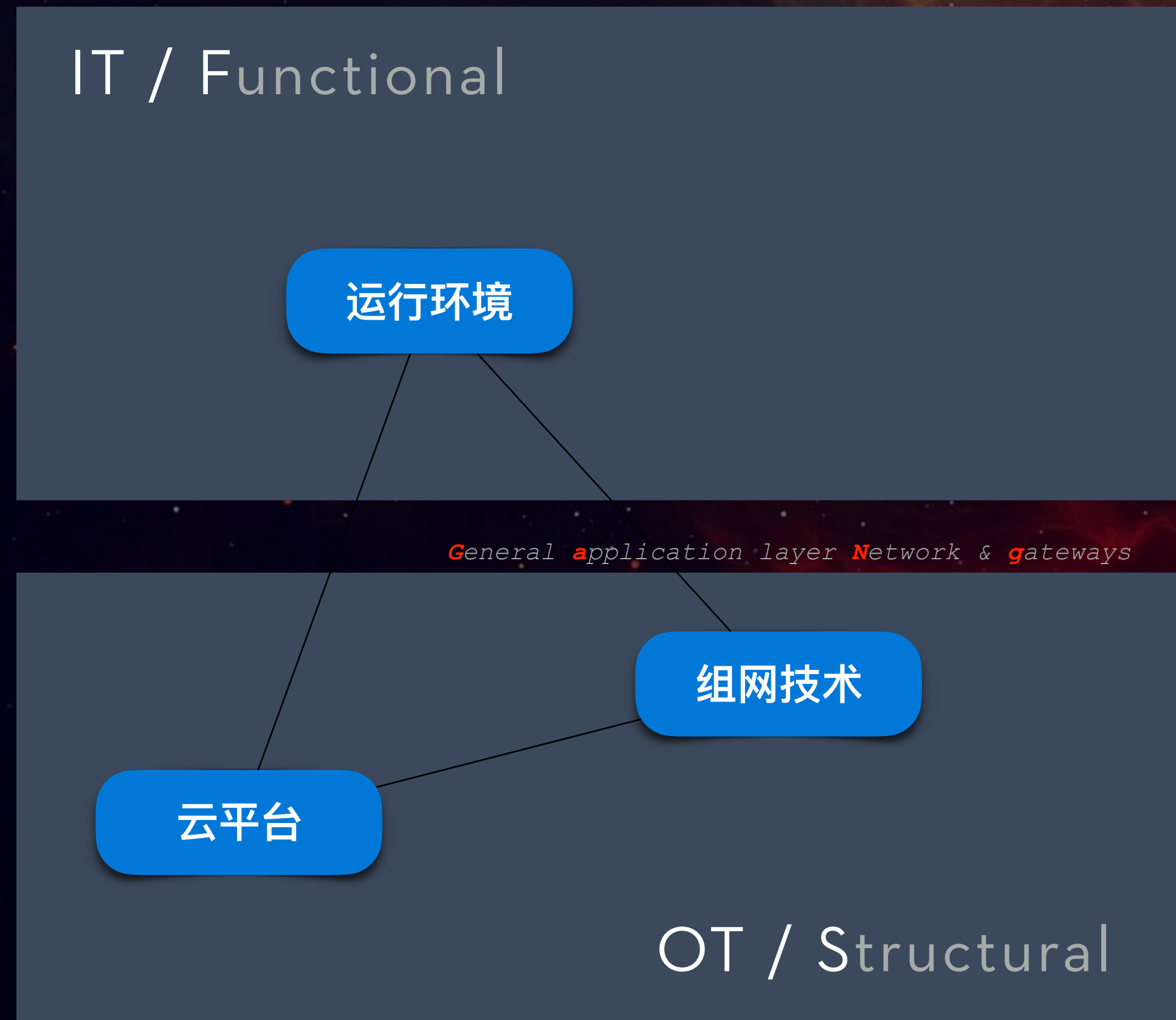
组一个网

宣告  
应用层网络

定位设备  
(模板/可视工具)

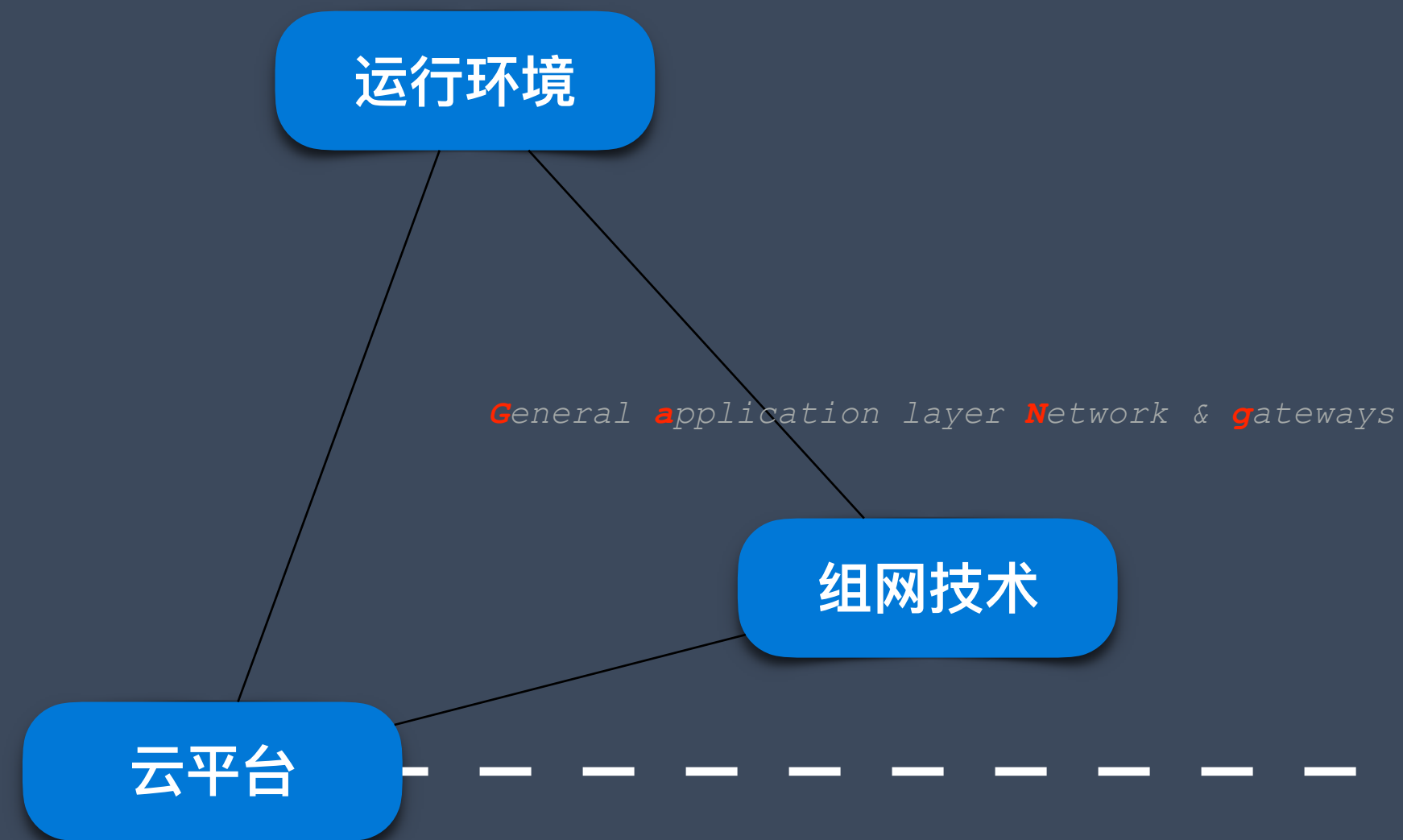
面向抽象环境  
开发应用

# Sluff - IoT Core Reference Architecture



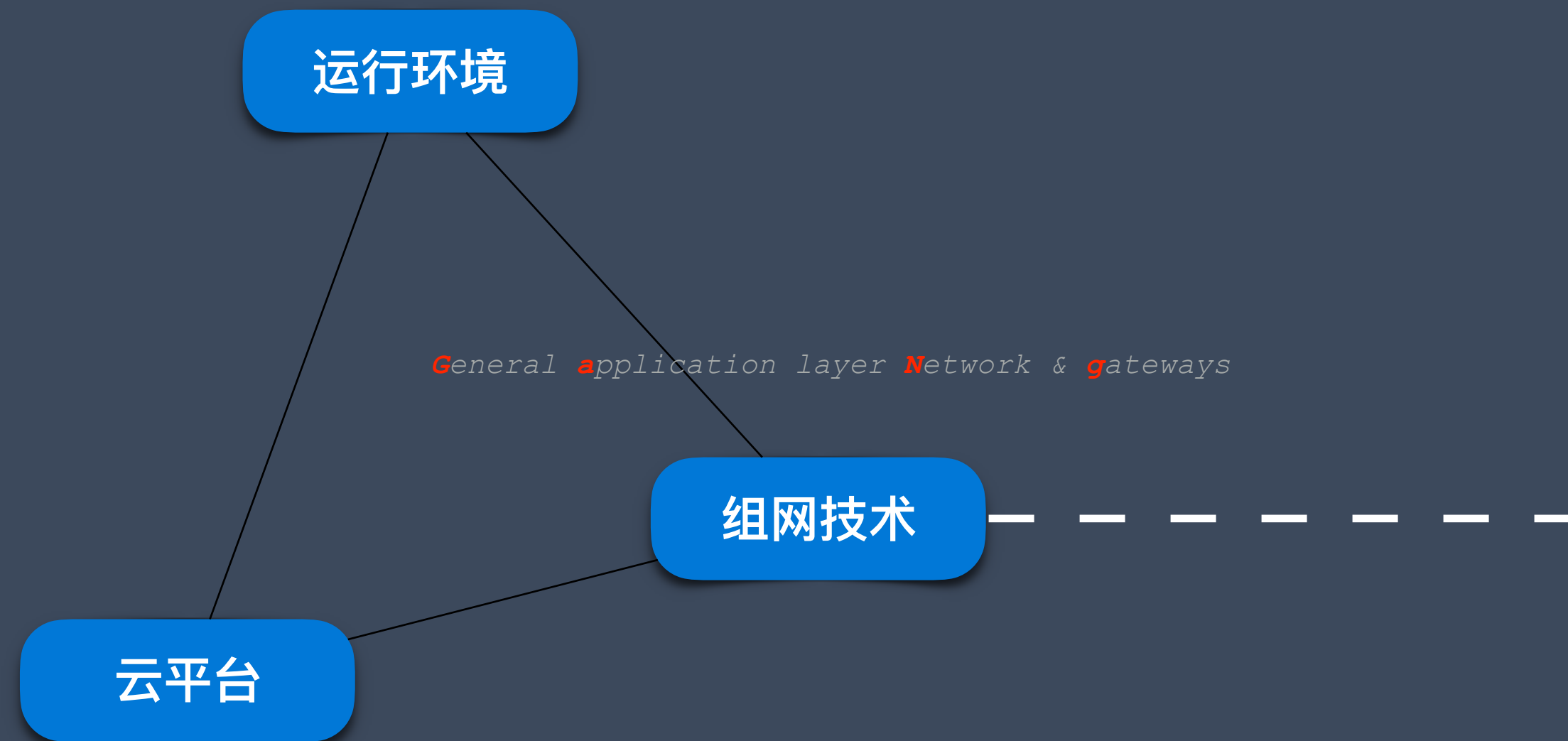
[sluff.io](http://sluff.io)  
[sluff.github.io](https://sluff.github.io)  
[bridgehosting.cn](http://bridgehosting.cn)

# Sluff - IoT Core Reference Architecture



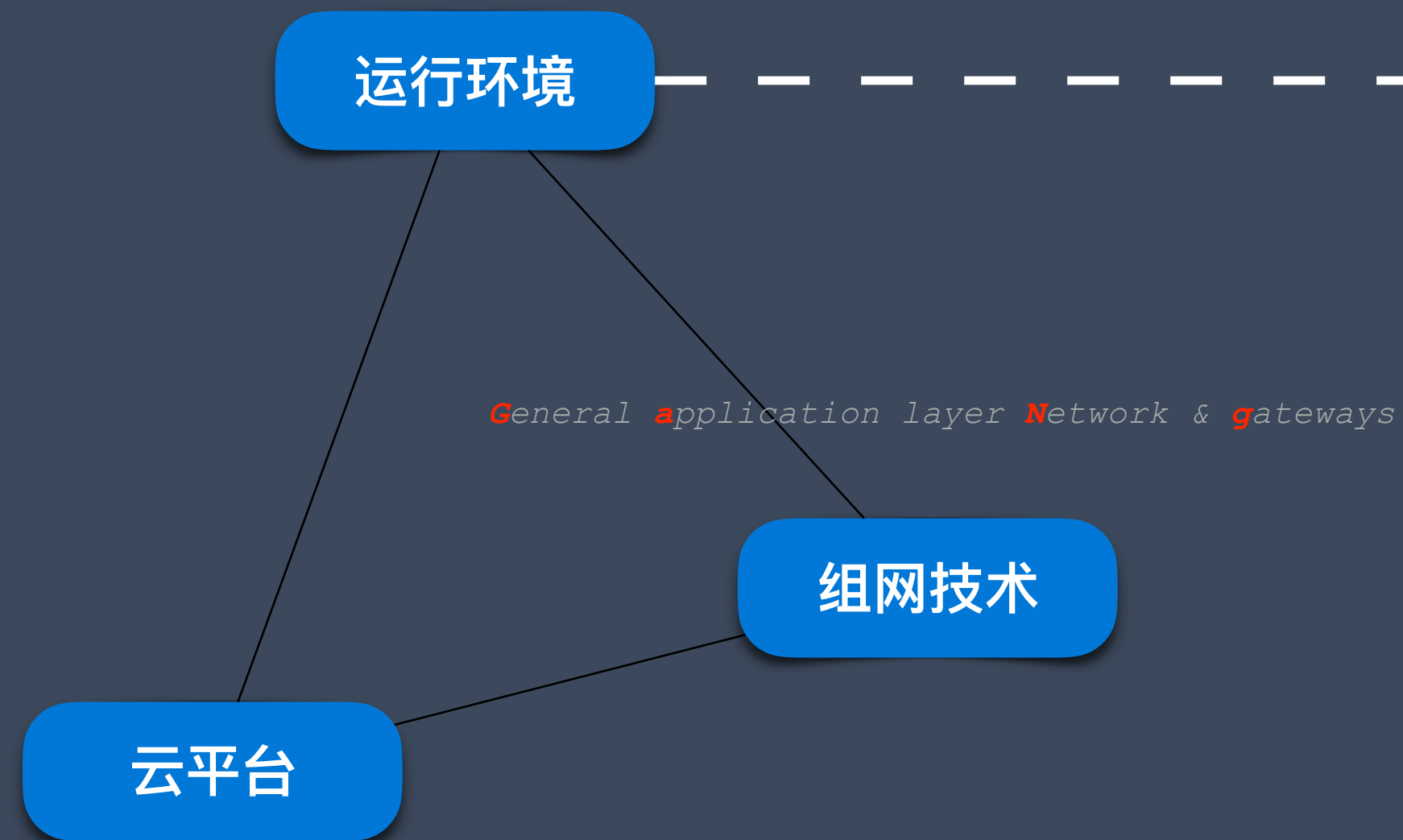
```
- - - - -> > uc init  
> npm install  
> npm start  
...  
> uc deploy  
> uc log
```

# Sluff - IoT Core Reference Architecture



```
// JSON Style, the facade  
// of network  
{  
  "id": "localhost",  
  "gateways": [{  
    ...  
  }],  
  
  "devices": [{  
    "id": "me",  
    "className": "Button"  
  }]  
}
```

# Sluff - IoT Core Reference Architecture



→ how to run me?

```
// JSON Style, the facade
// of network
```

```
{
  "id": "localhost",
  ...
  "devices": [{
    "id": "me",
    "className": "Button"
  }]
}
```

```
// templet
{
  btn: "#me",
  box: {
    led: "#led"
    ...
  }
  ...
}
```

```
// environment
```

```
{
  btn: {
    state: ...,
    click: ...
  },
  box: {
    led: {
      text: ...,
      ...
    },
    ...
  }
  ...
}
```



```
// 面向抽象环境开发应用
```

```
cloud.ready.then(({btn, box}) => {
  btn.click();
  ...
})
```



```
cloud.ready  
  .then (main.go)
```





# Sluff: Thing是可被组织的元素



组网技术: 设备连入、通讯等, 通讯级别的安全性

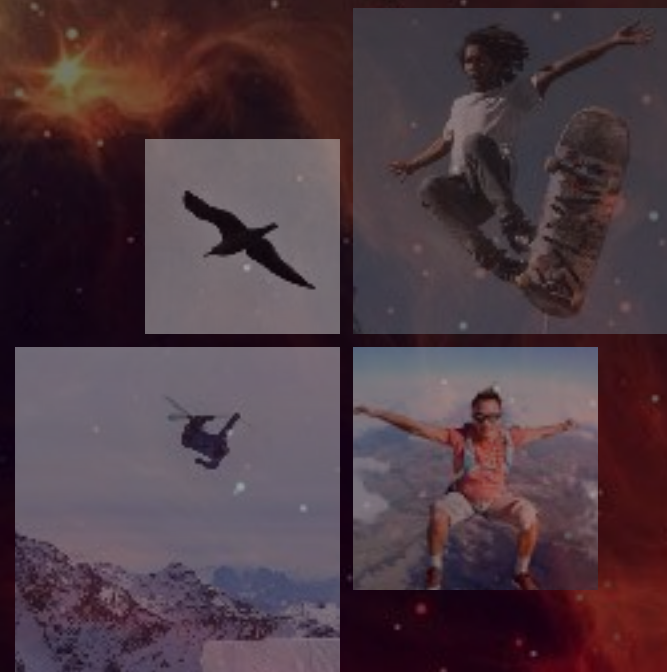


运行环境: 功能与可用性



云平台: 安全性、可靠性与第三方

Handwritten signature or mark in the bottom right corner.



sluff.io

A handwritten signature in the bottom right corner of the image.