

# 微博视频转码系统架构演进

新浪微博 - 杜东澄

# 概要

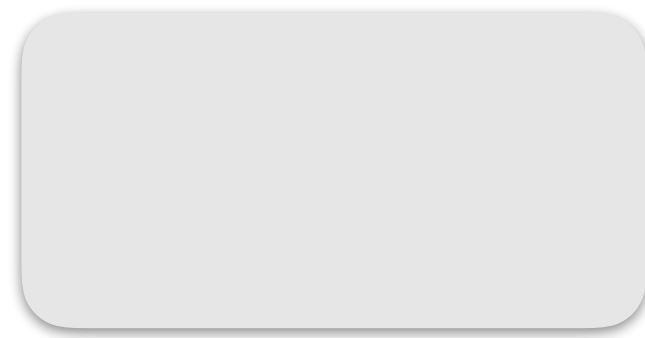
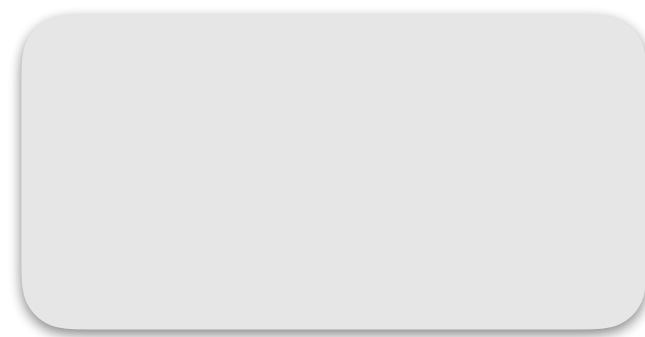
- 微博转码系统的特点
- 新老系统架构演进
- DAG 框架
- 应用场景

# 微博转码系统特点

- 数百万转码任务量
- 多格式、多编码、多分辨率下发
- 多种业务类型：视频转码、截图、抽帧、gif2video、全景截图等

# 旧的转码系统

调度器

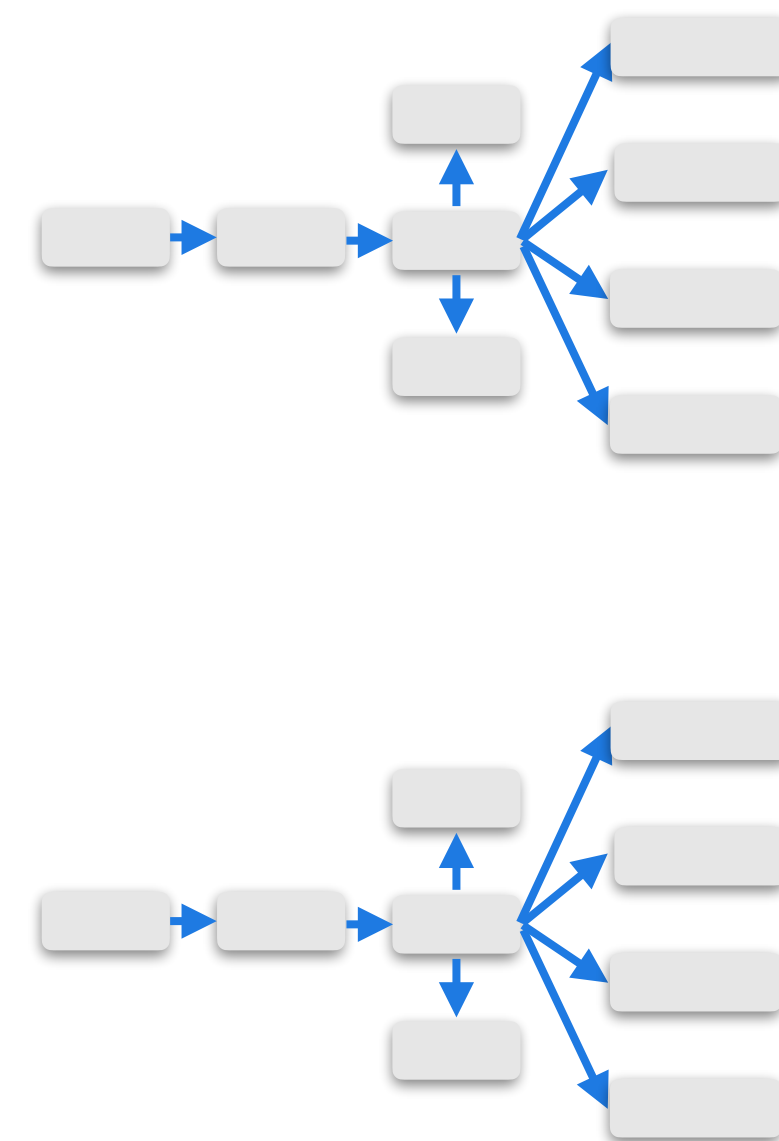


执行器

# 旧的转码系统

调度器

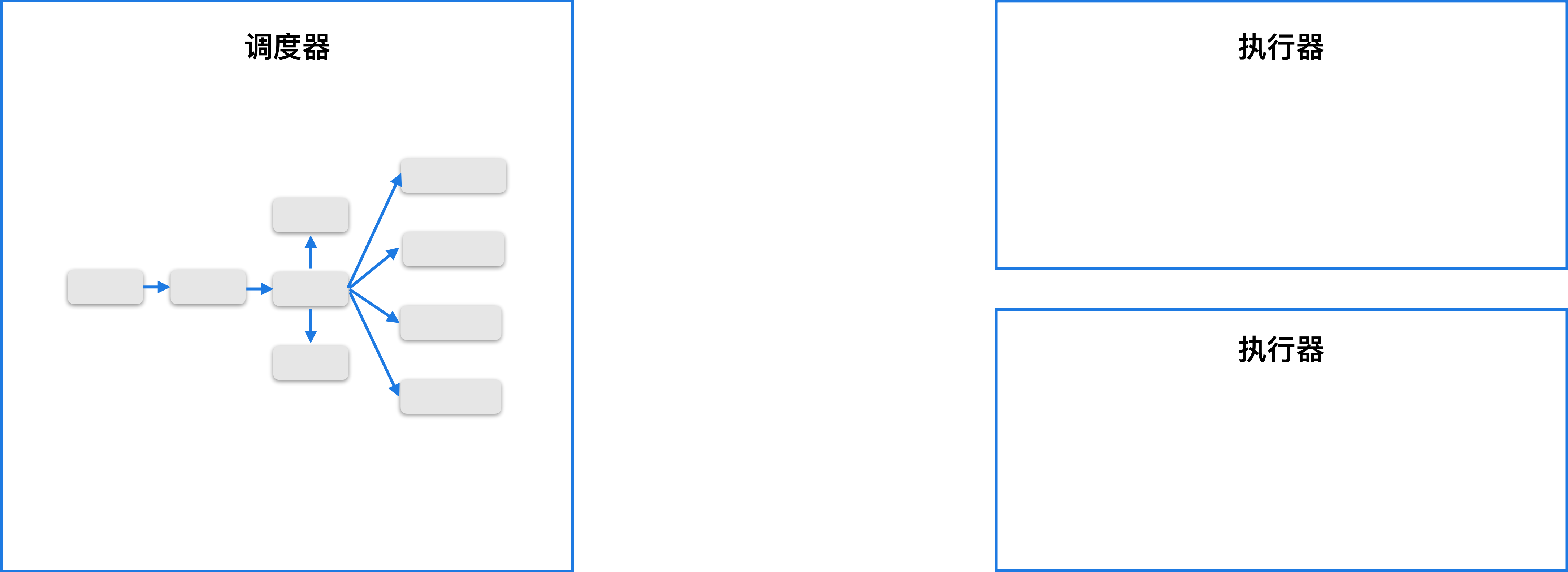
执行器



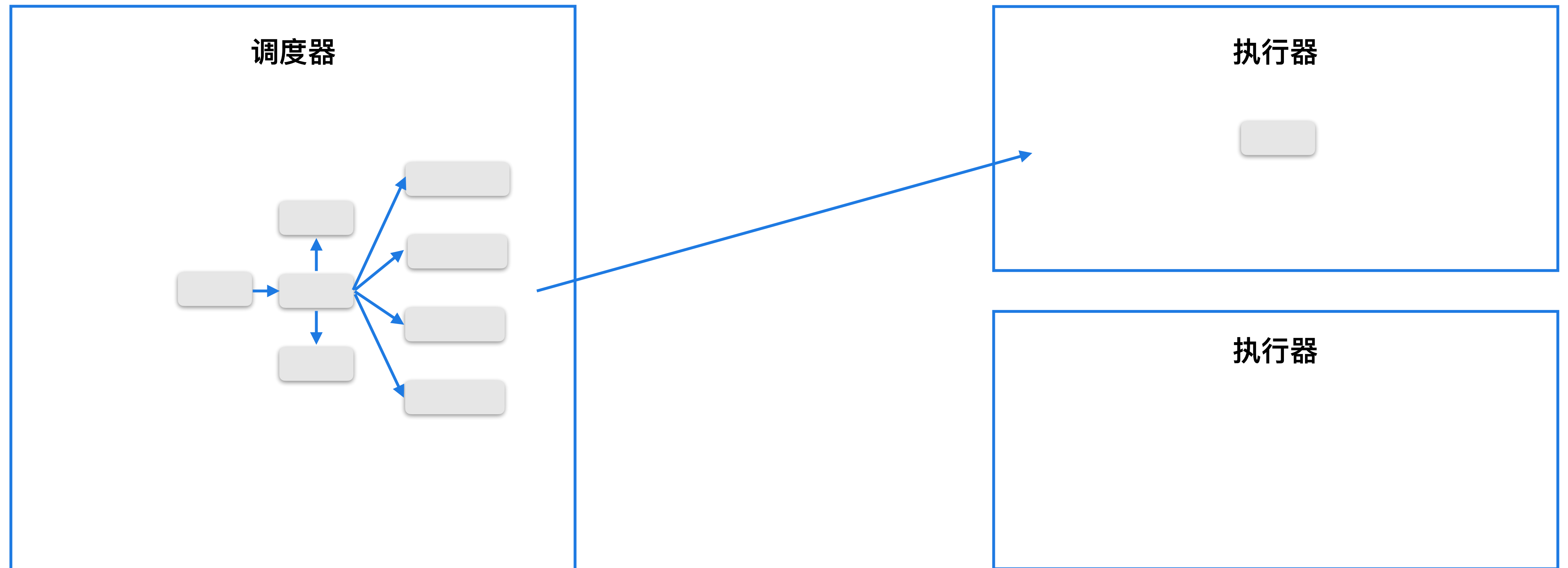
# 暴露的问题

- 不易扩展
- 粗粒度调度
- 缺少任务监控

# 设想新的转码系统

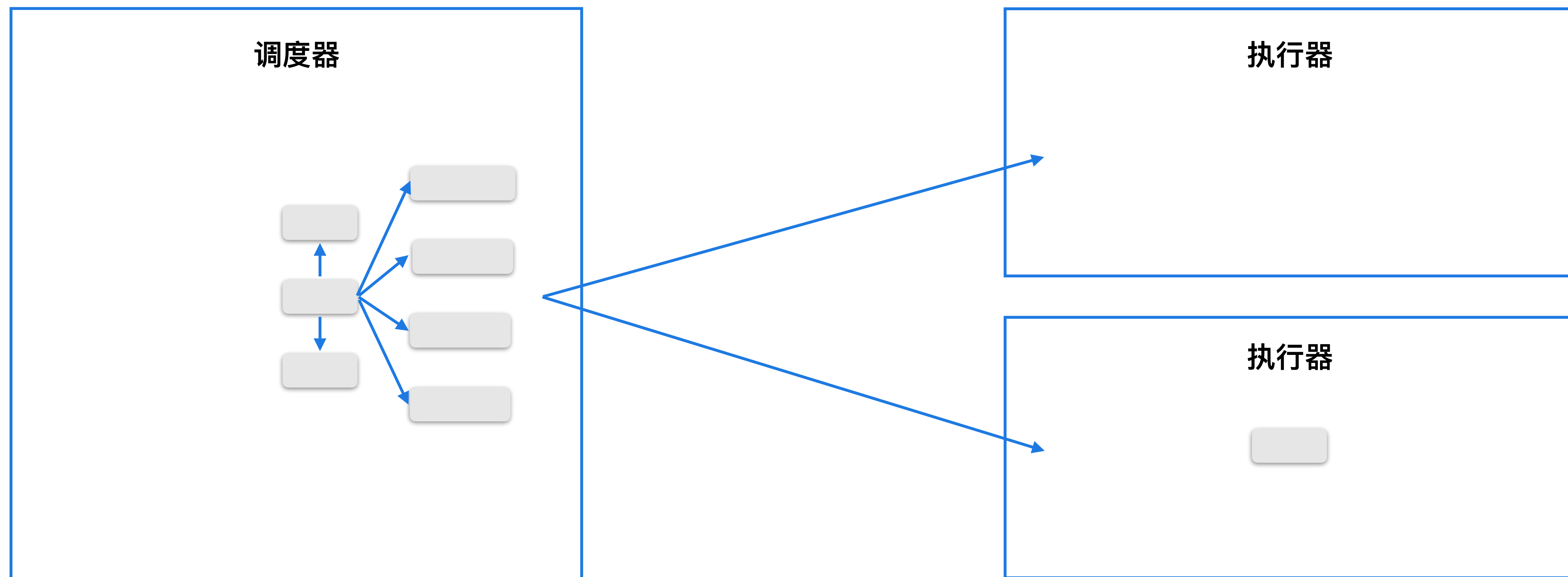


# 设想新的转码系统

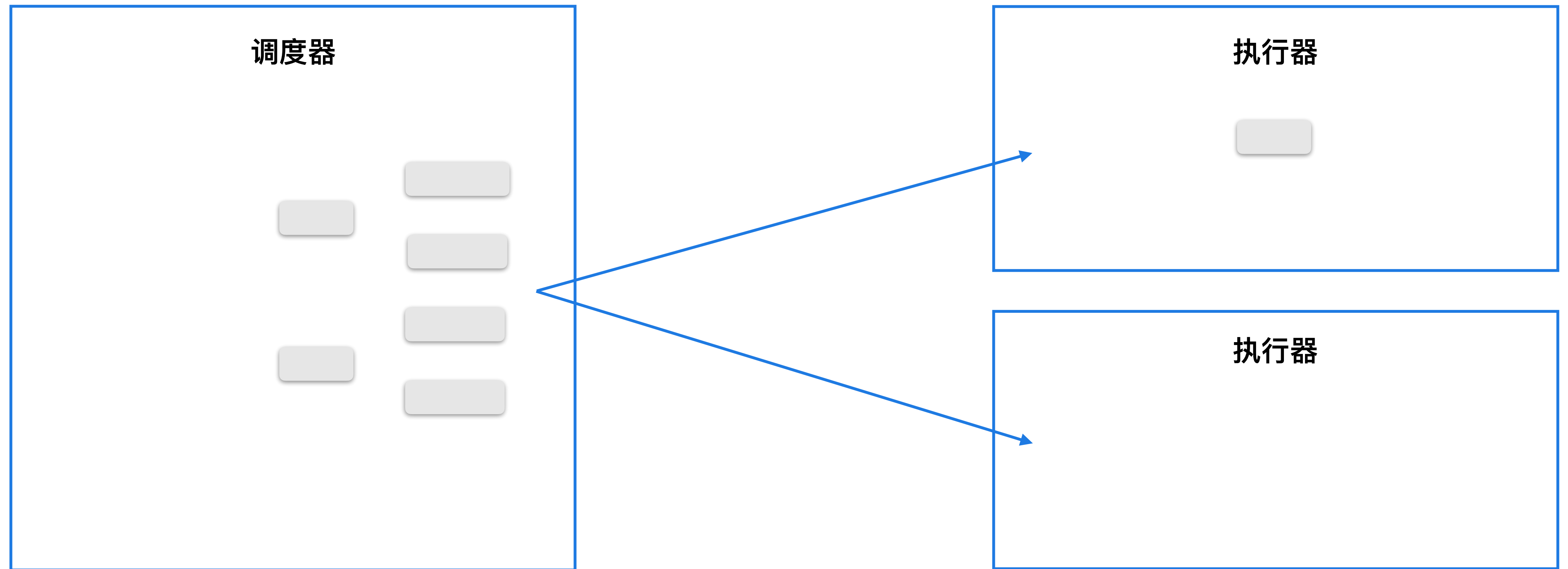




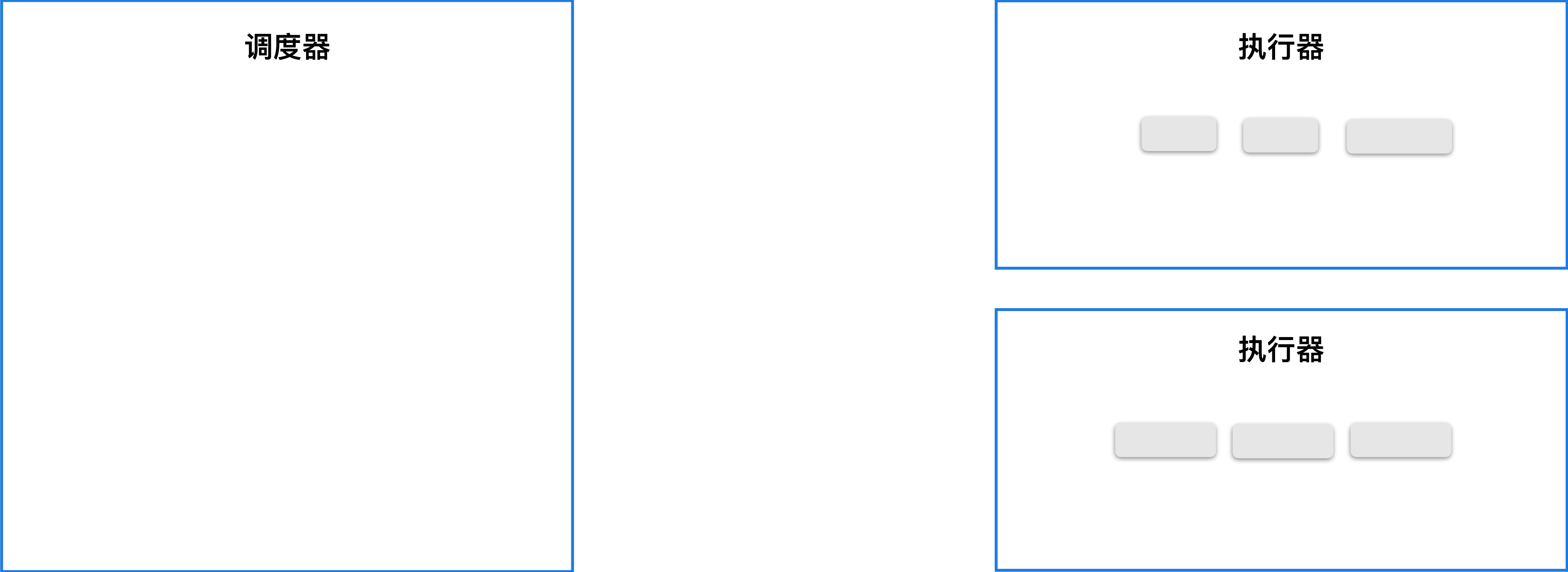
# 设想新的转码系统



# 设想新的转码系统



# 设想新的转码系统



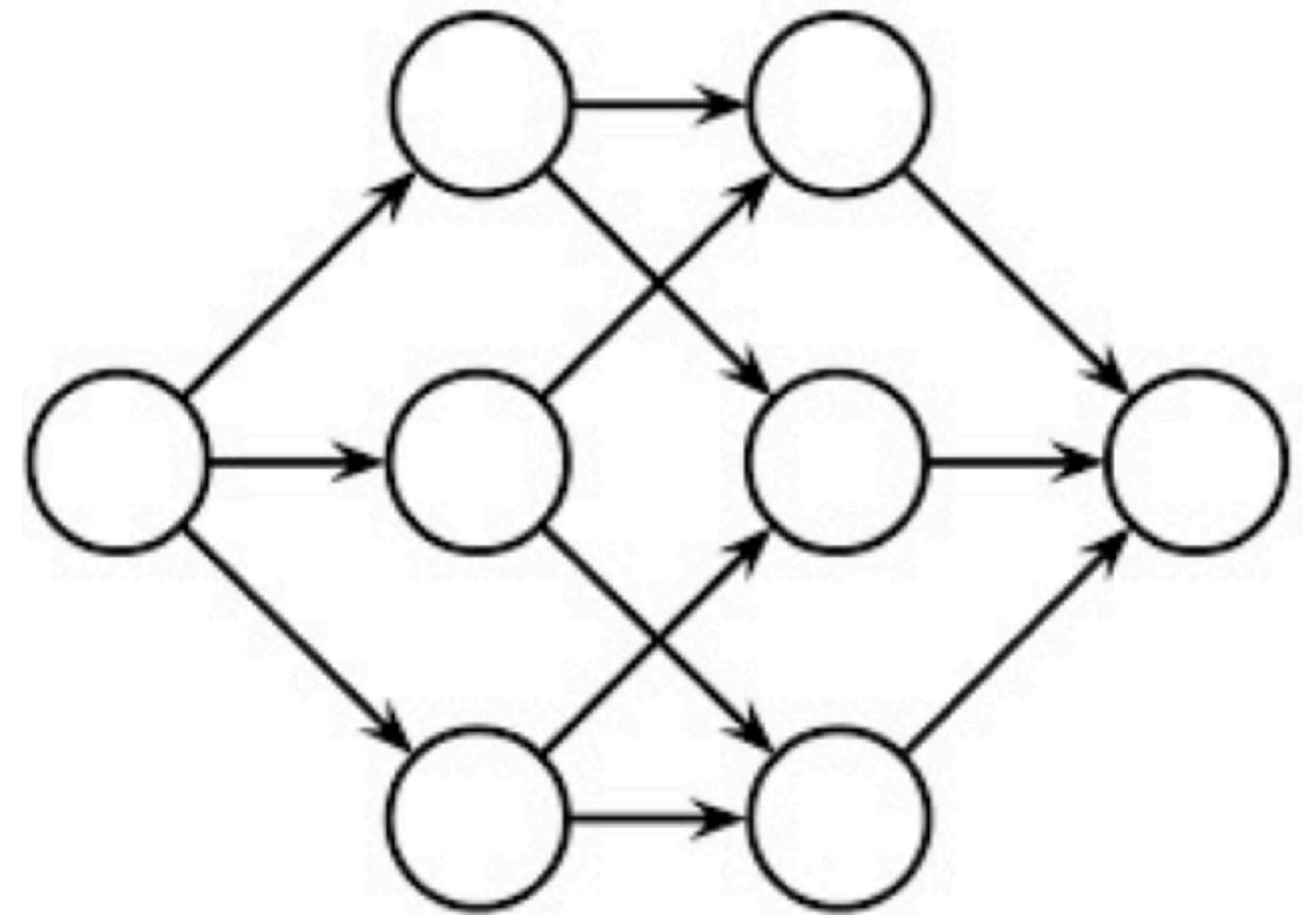
# 架构演进的目标

- 灵活的应对需求
- 快速的任务调度
- 稳定的服务架构

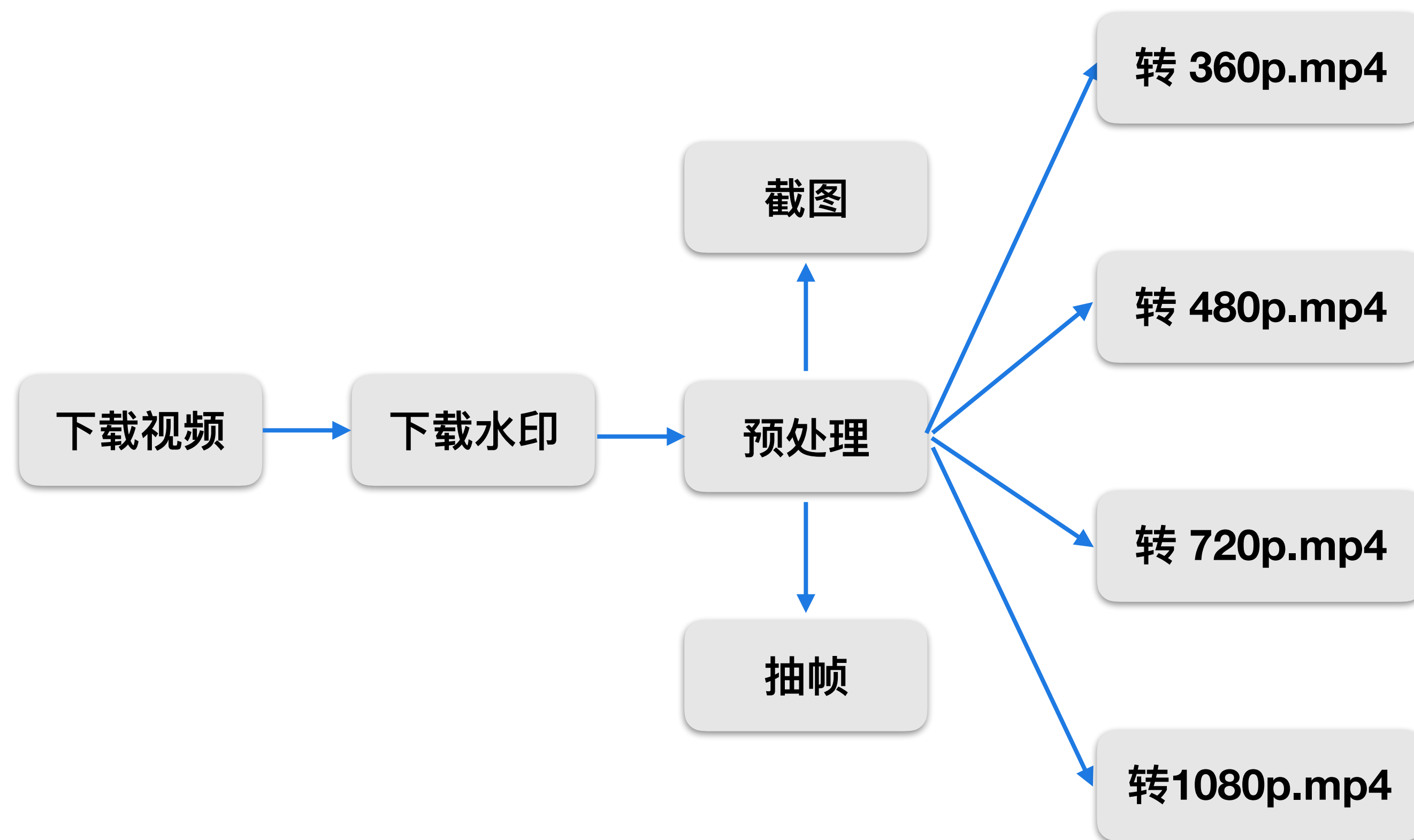


# 什么是DAG

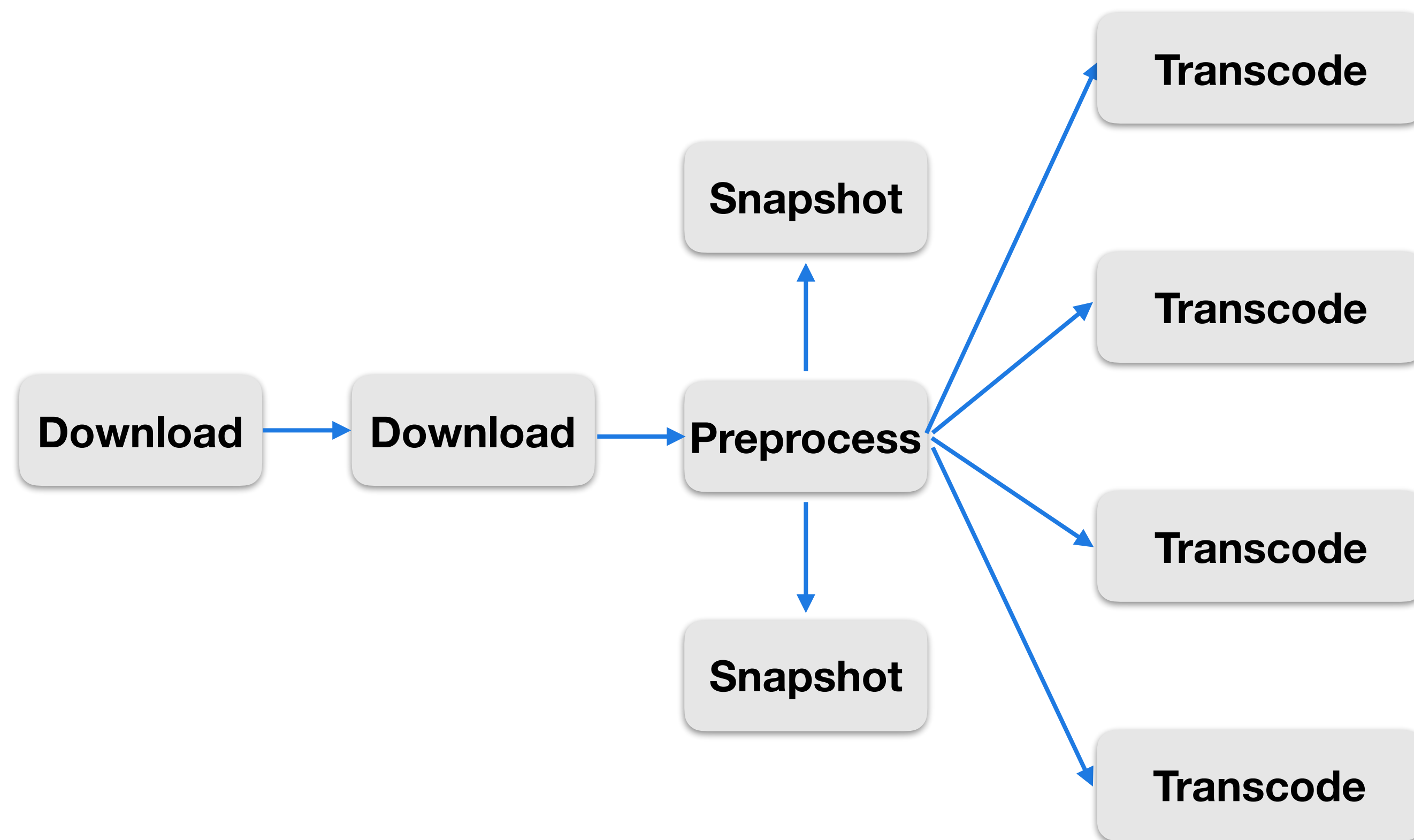
- 有向无环图 (Directed Acyclic Graph)
- 从任意顶点出发无法经过若干条边回到该点



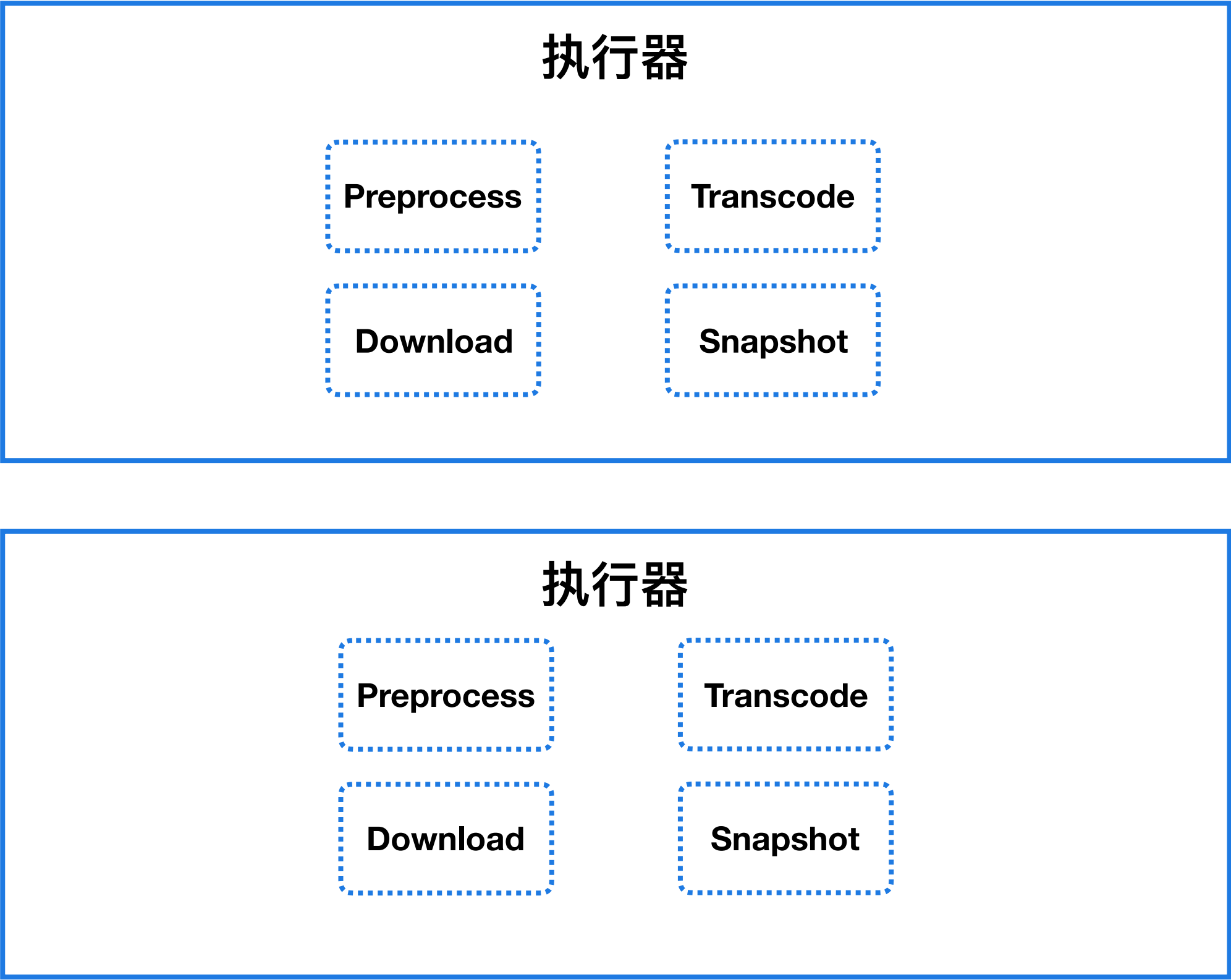
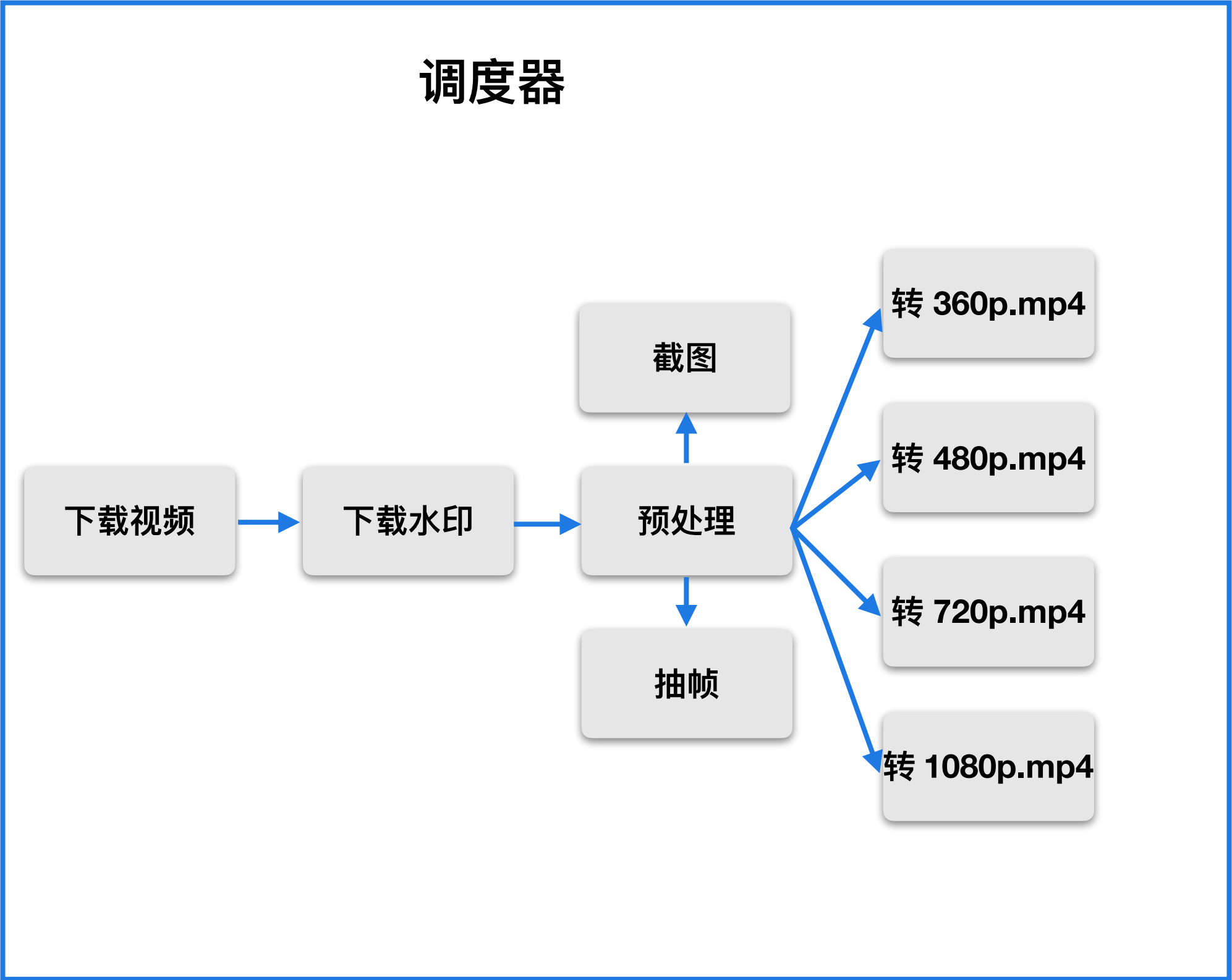
# 转码的业务逻辑



# 转码的DAG描述

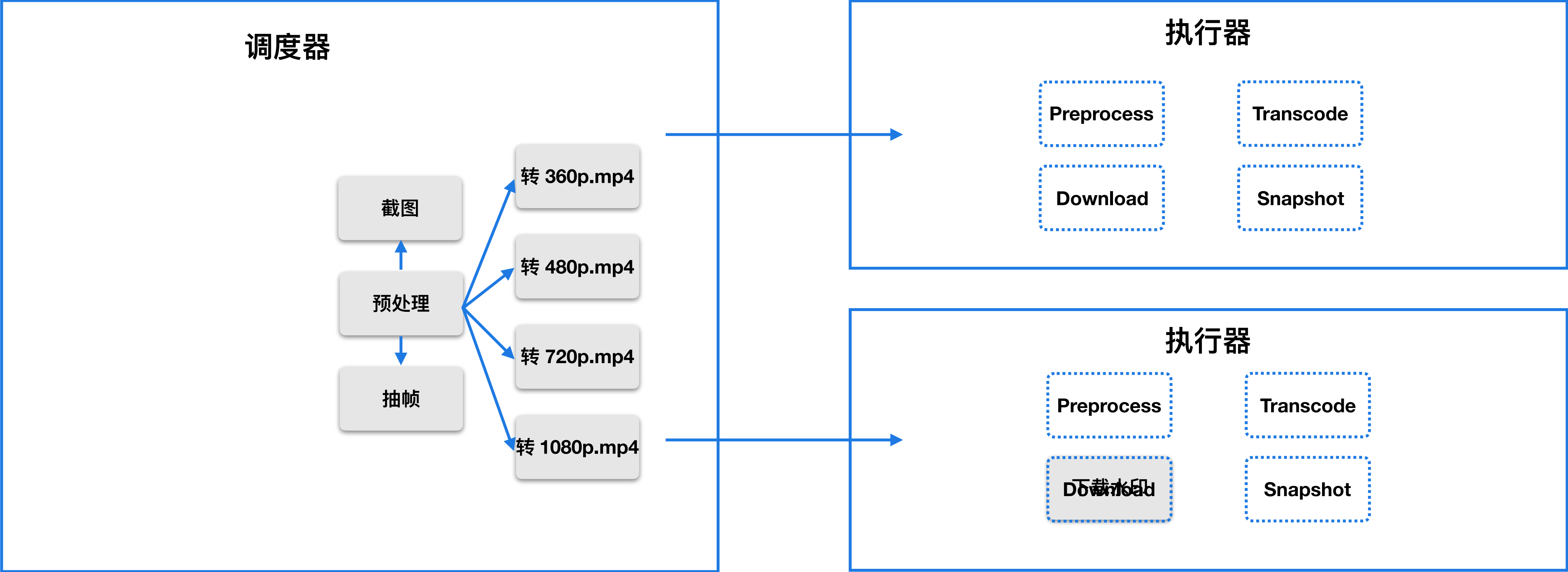


# DAG 系统运行

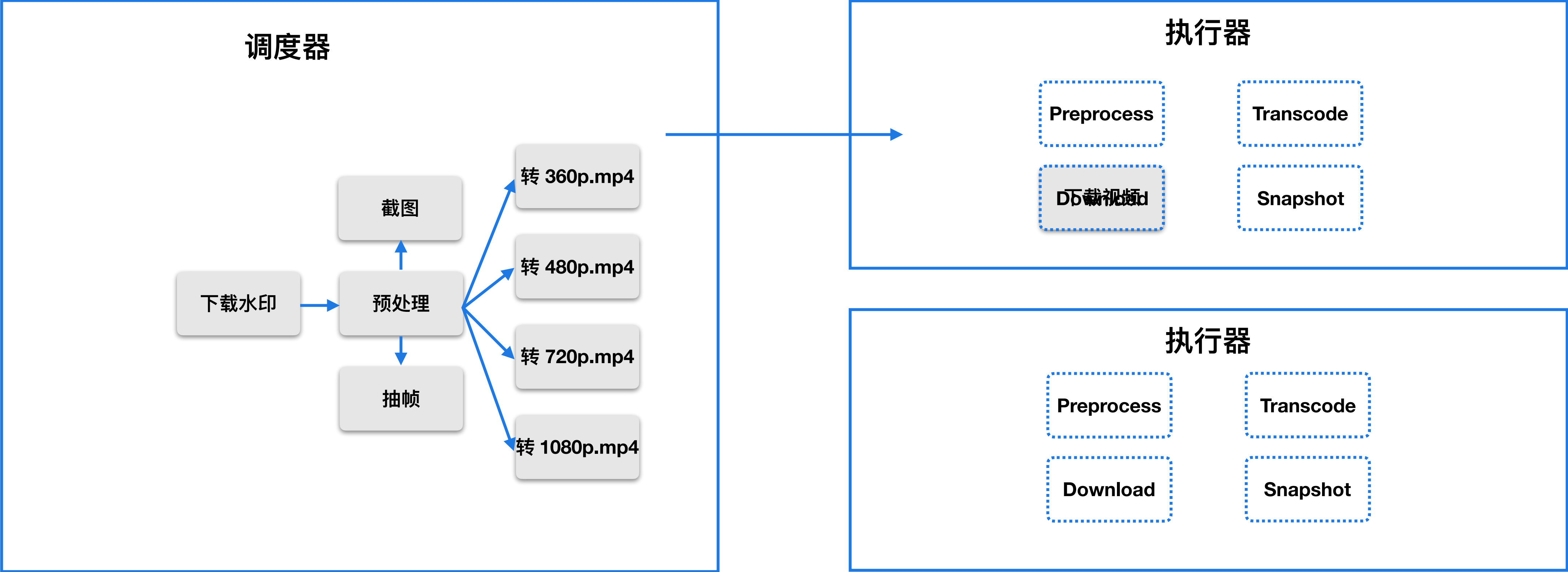




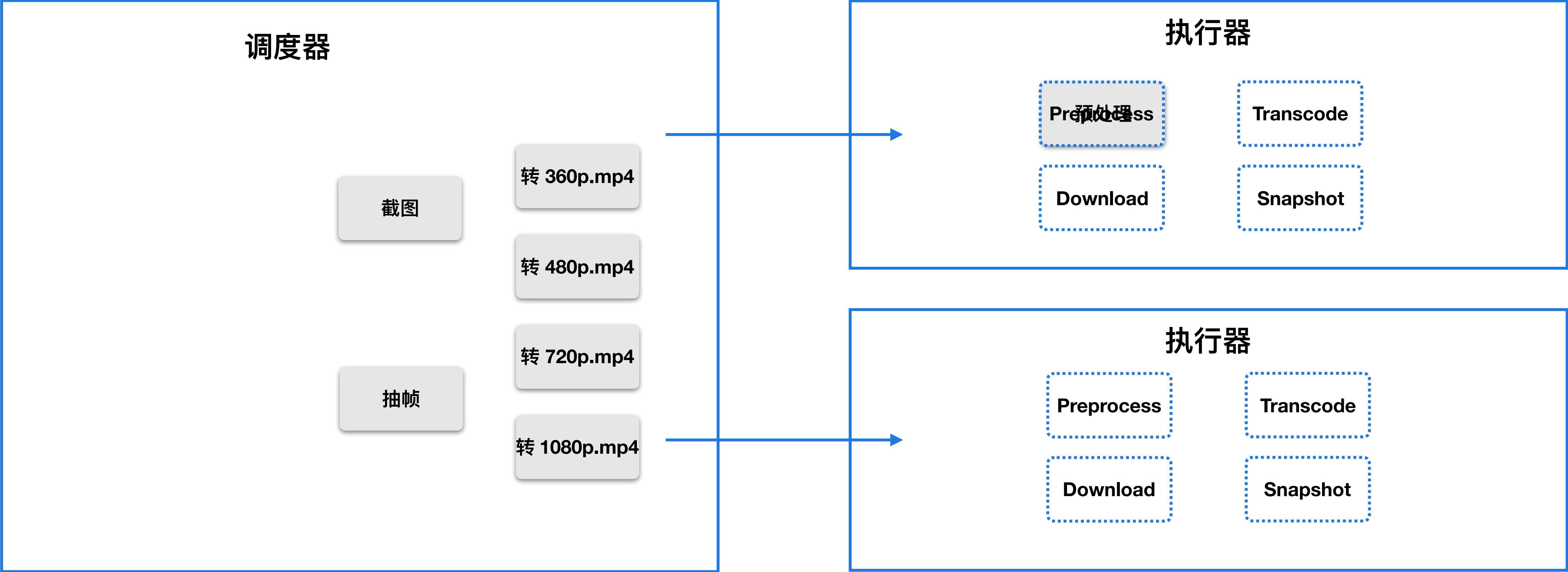
# DAG 系统运行



# DAG 系统运行



# DAG 系统运行



# DAG 系统运行

调度器

执行器

Preprocess

转码  
转1080p  
转360p

Download

抽帧  
Snapshot

执行器

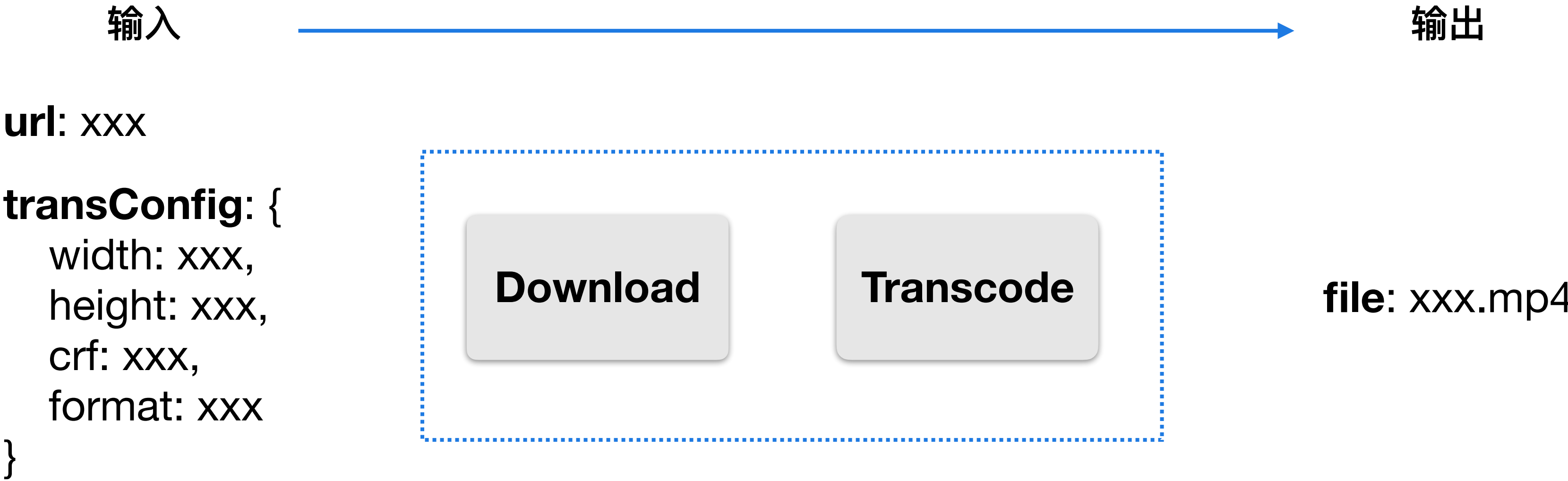
Preprocess

转码  
转1080p  
转720p

Download

Snapshot

# 实现：任务描述



# 实现：任务描述

```
task {  
  name 'download-input'  
  type 'Download'  
  input {  
    url config.url  
  }  
  output { it->  
    context.inputVideo = it.file  
  }  
  next 'transcode'  
}
```

```
task {  
  name 'transcode'  
  type 'Transcode'  
  input {  
    input context.inputVideo  
    config config.transConfig  
  }  
  output { it->  
    context.file = it.outputVideo  
  }  
}
```

# 应用场景

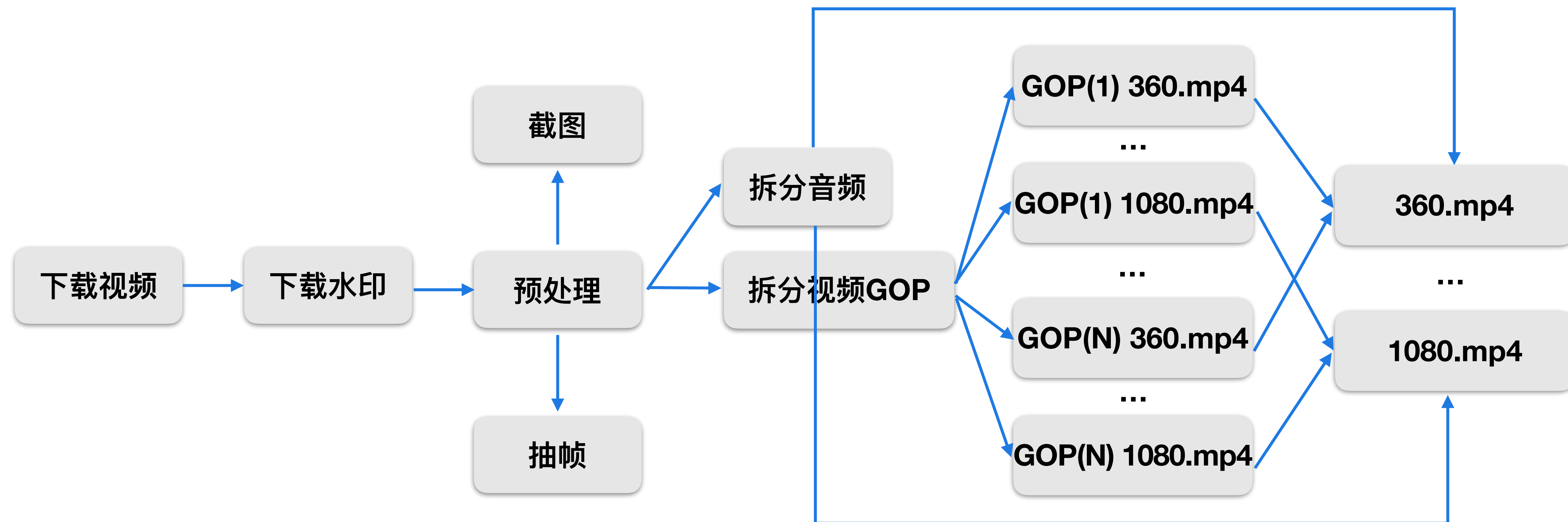
- 分片转码
- DASH
- 监控

# 场景： 分片转码

- 拆分音视频轨
- 将视频轨拆分成多个 GOP 分片
- 提高大视频的并发处理速度



# 场景：分片转码

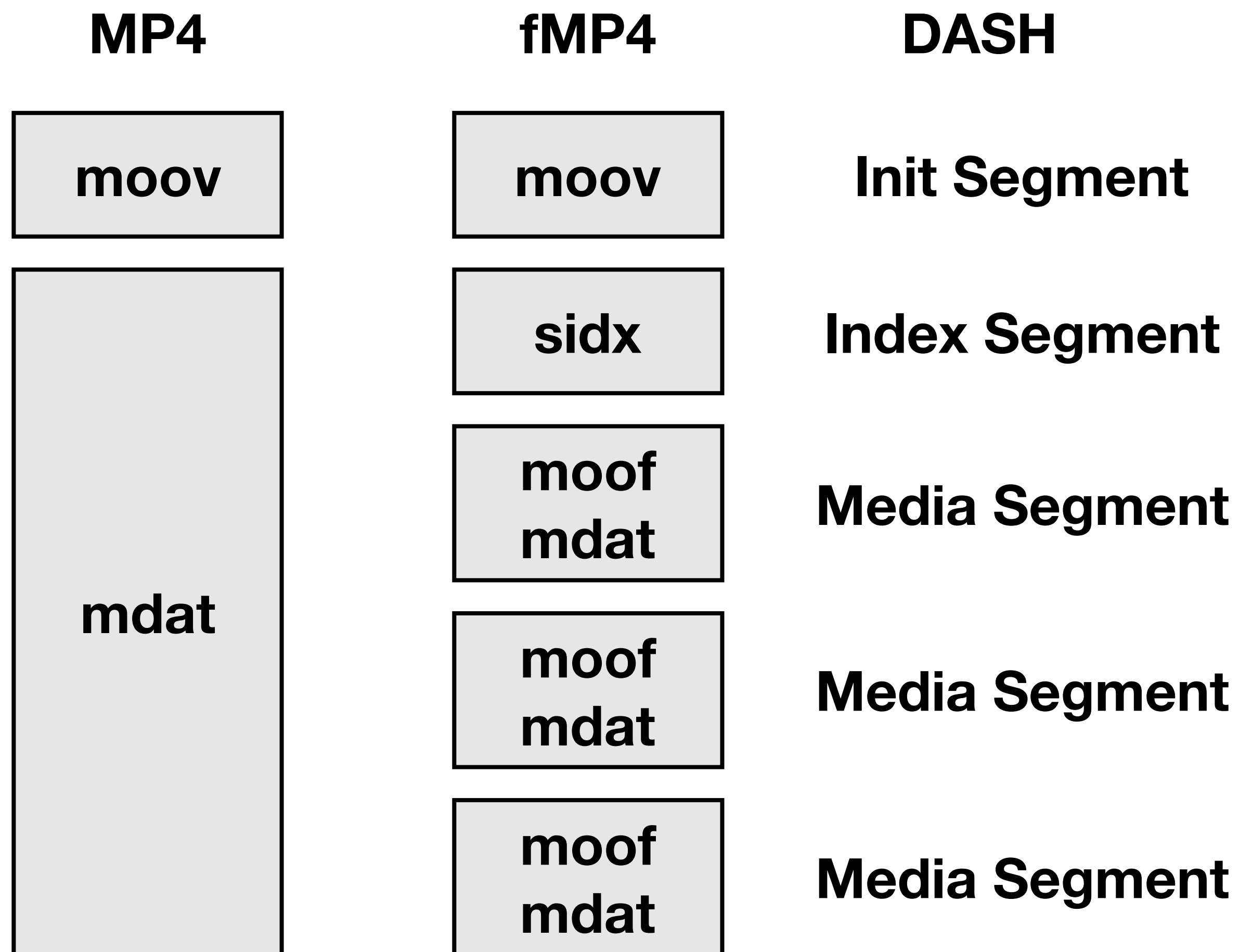


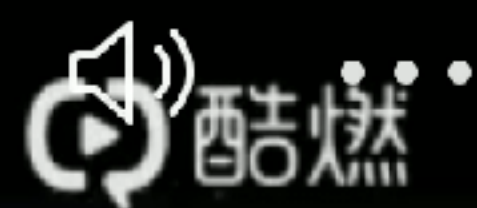
# 场景： DASH转码

- 为什么要做 DASH?
  - 多码率下发
  - 无缝切换清晰度
  - 播放码率自适应

# 场景：DASH转码

- 怎么做?
- 音视轨分离
- fMP4





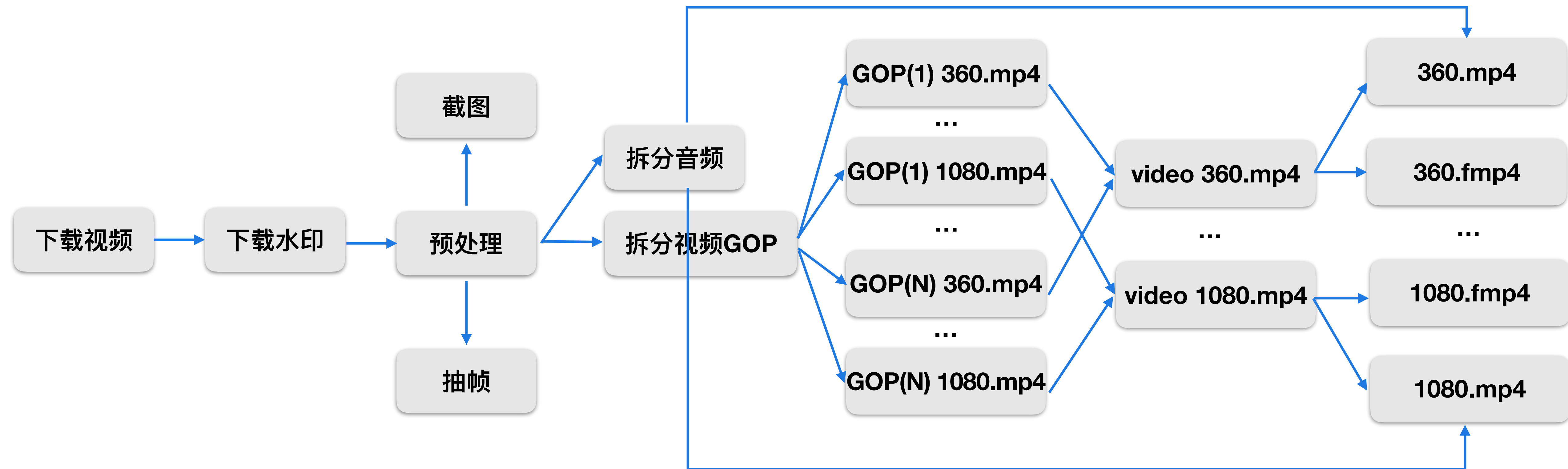
正在切换到**高清**，请稍候...

然后开始了他真正的计划  
and then began his real plan.

已成功切换高清

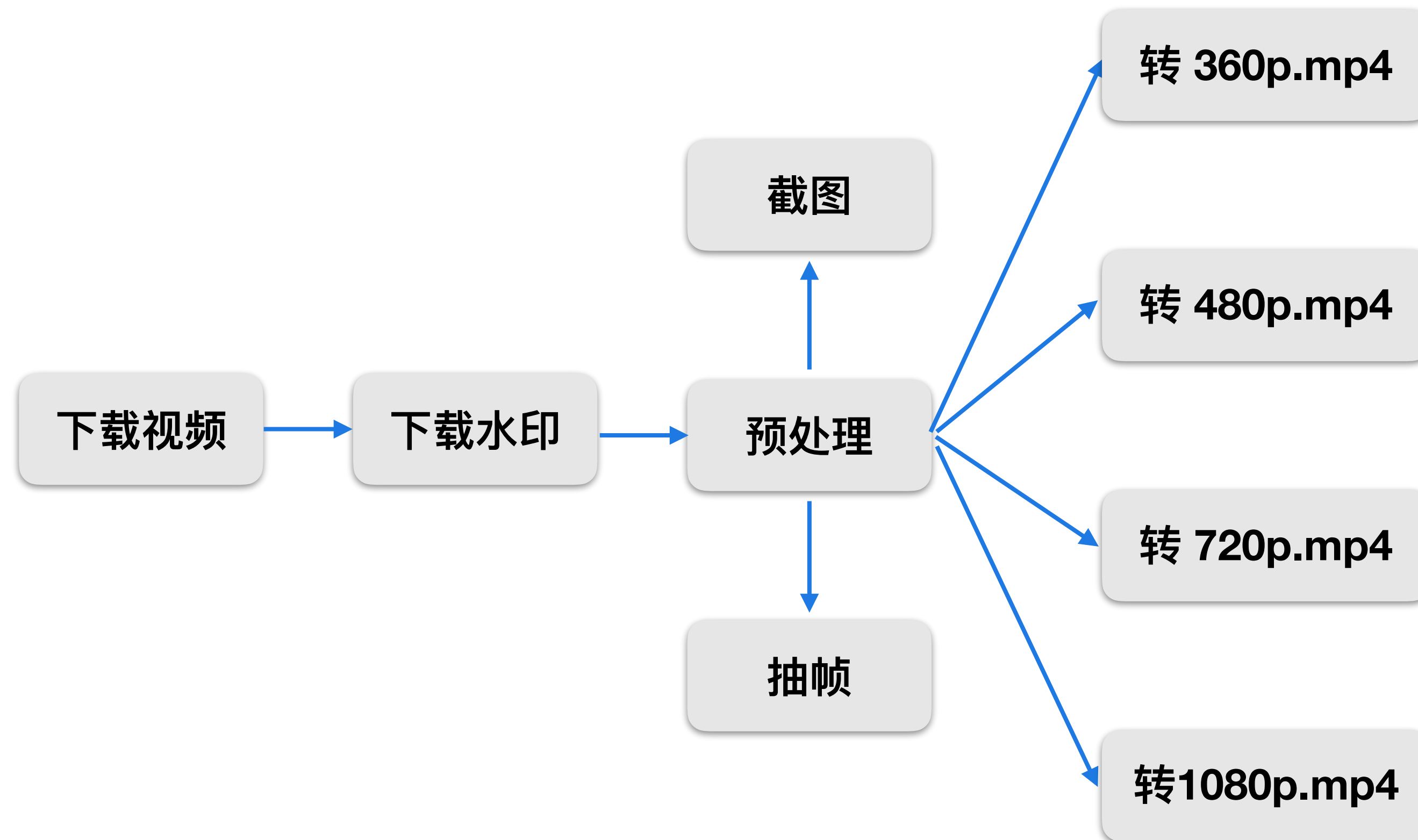
什么  
What?

# 场景：DASH转码



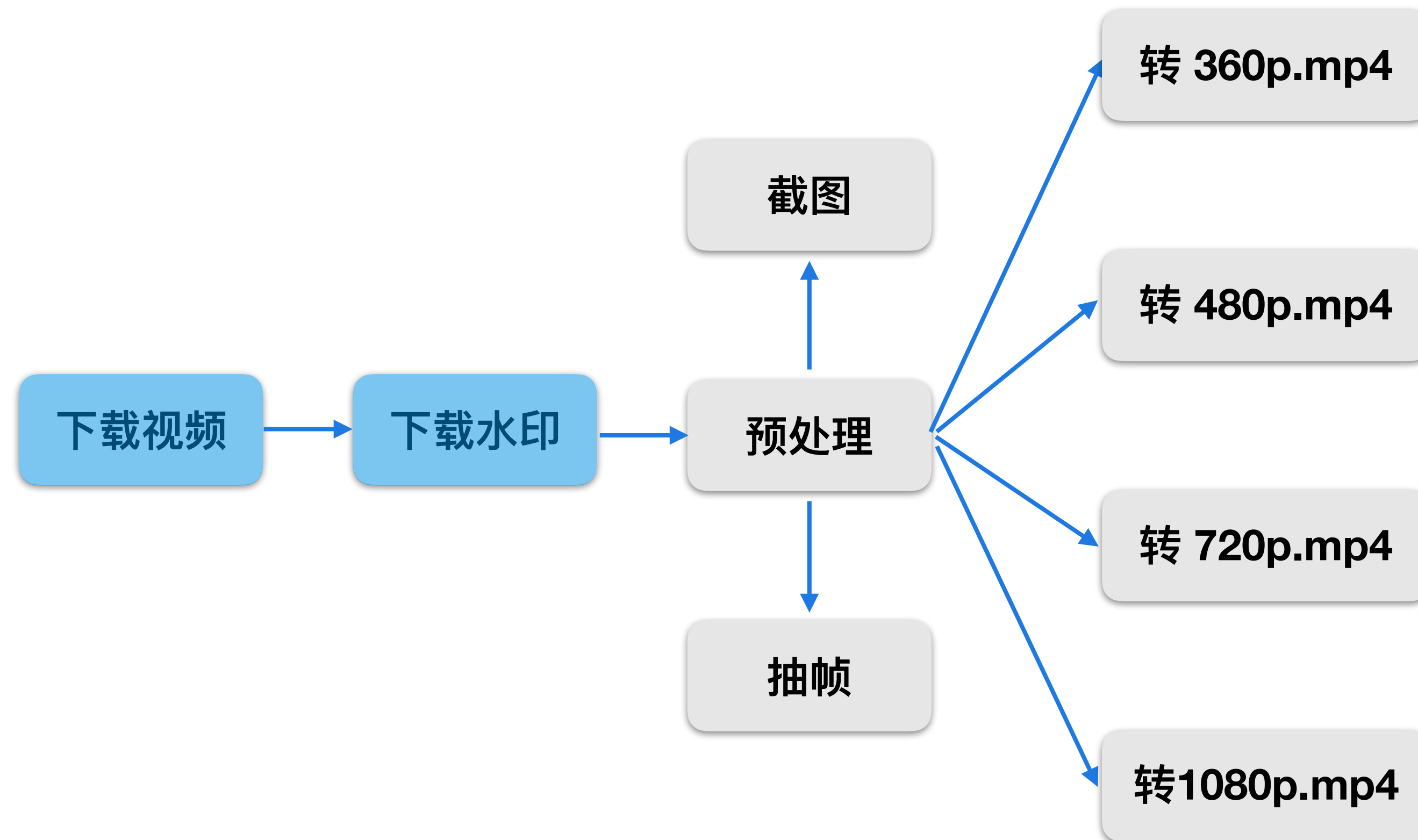
# 场景：任务监控

0%



# 场景：任务监控

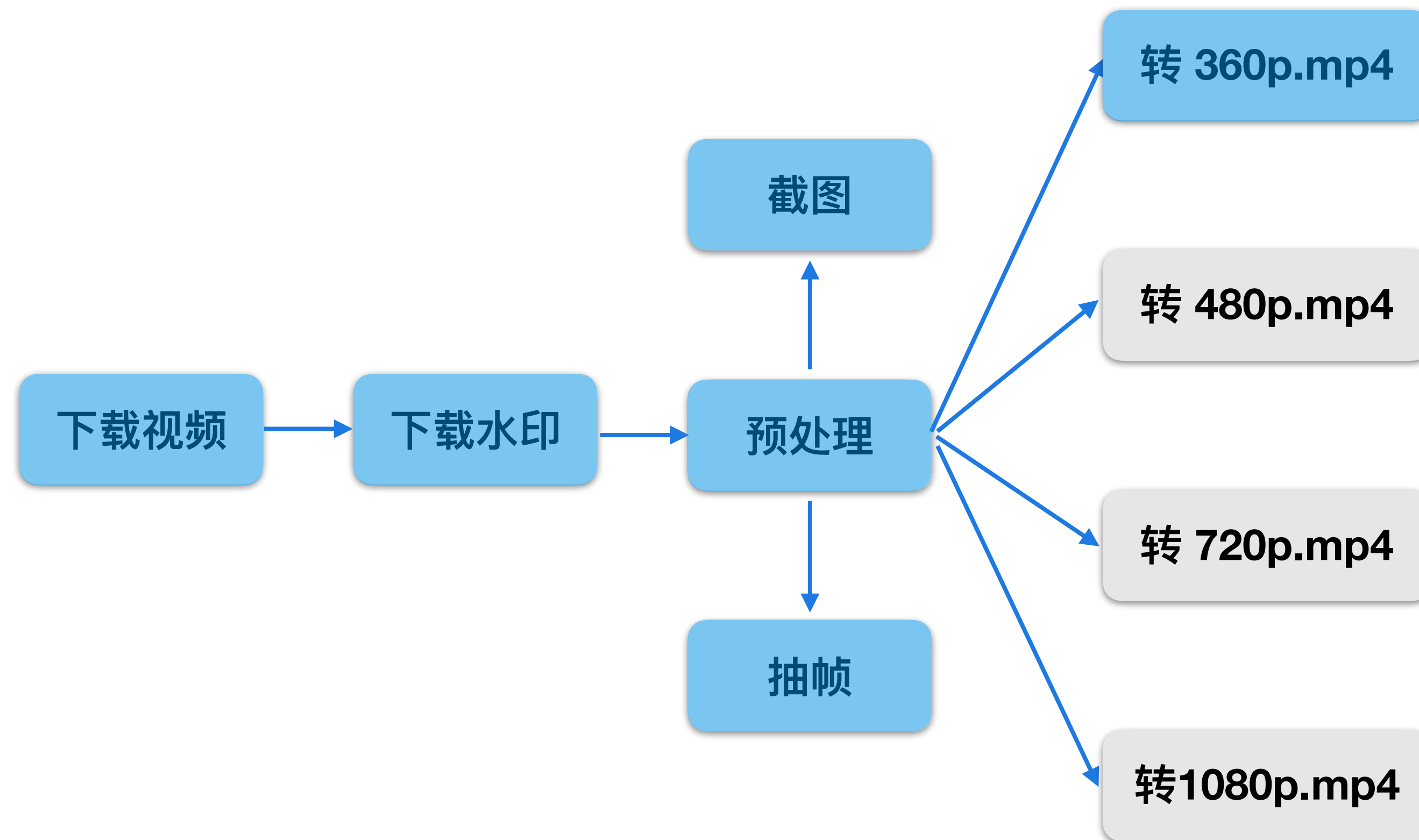
20%





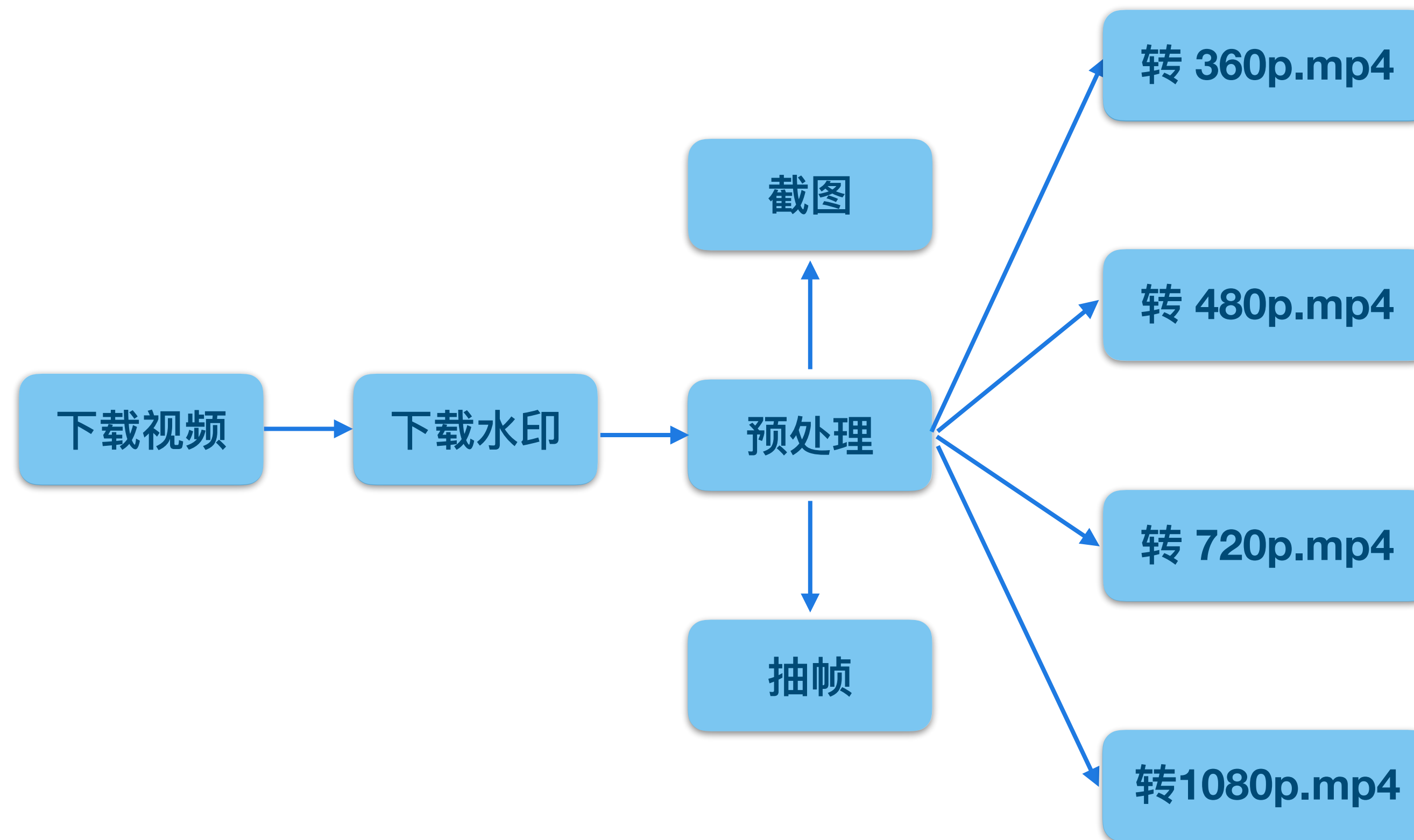
# 场景：任务监控

60%

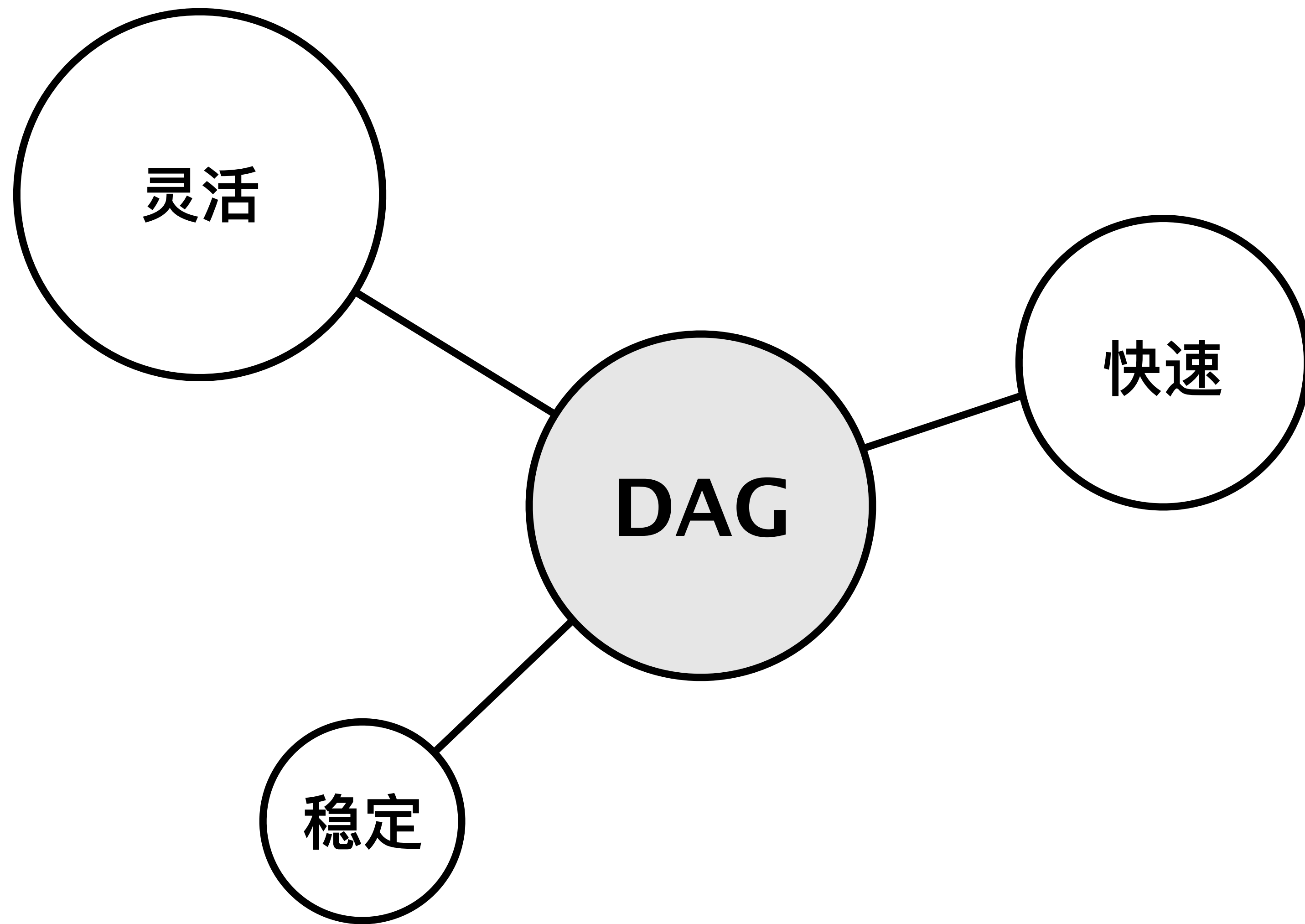


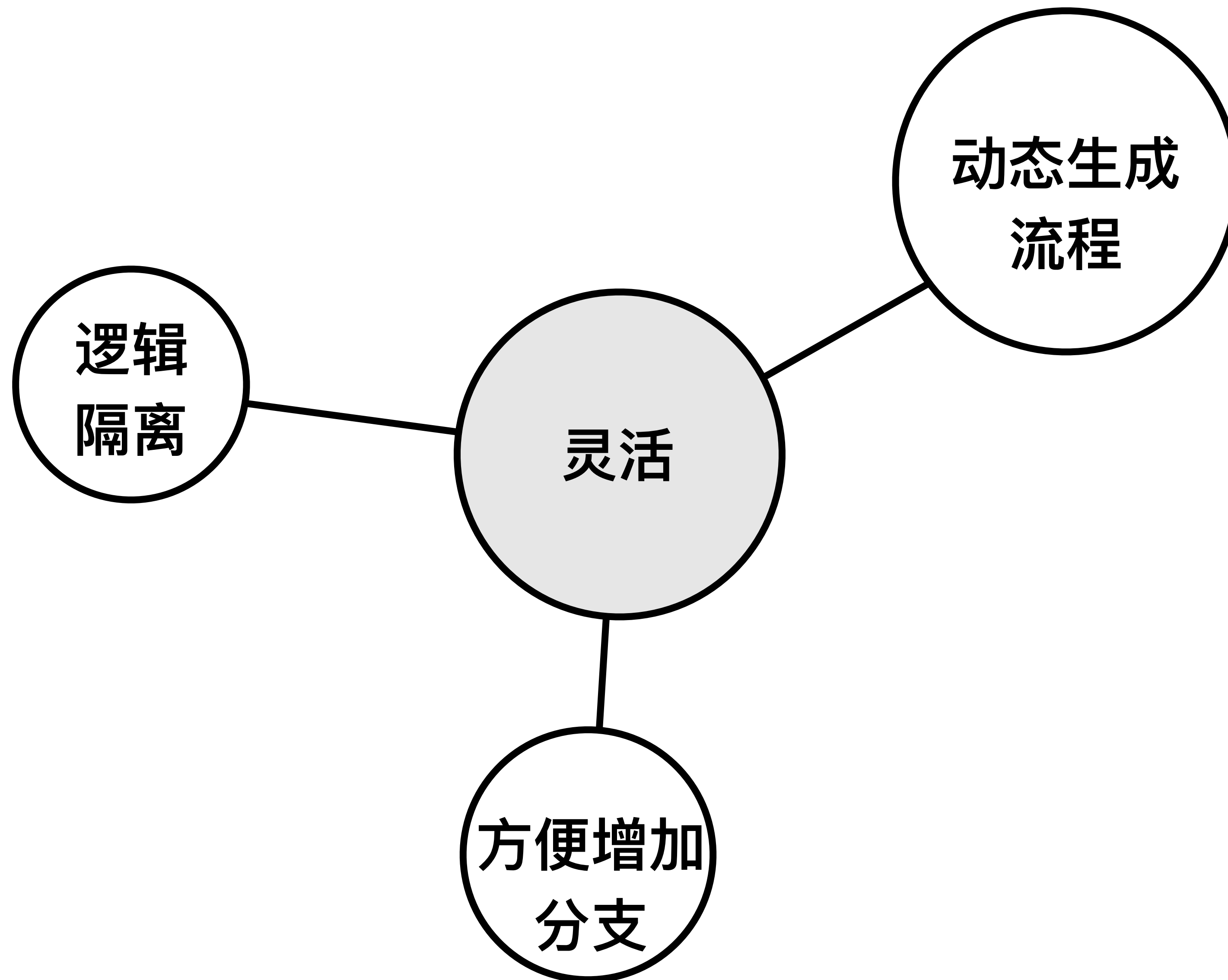
# 场景：任务监控

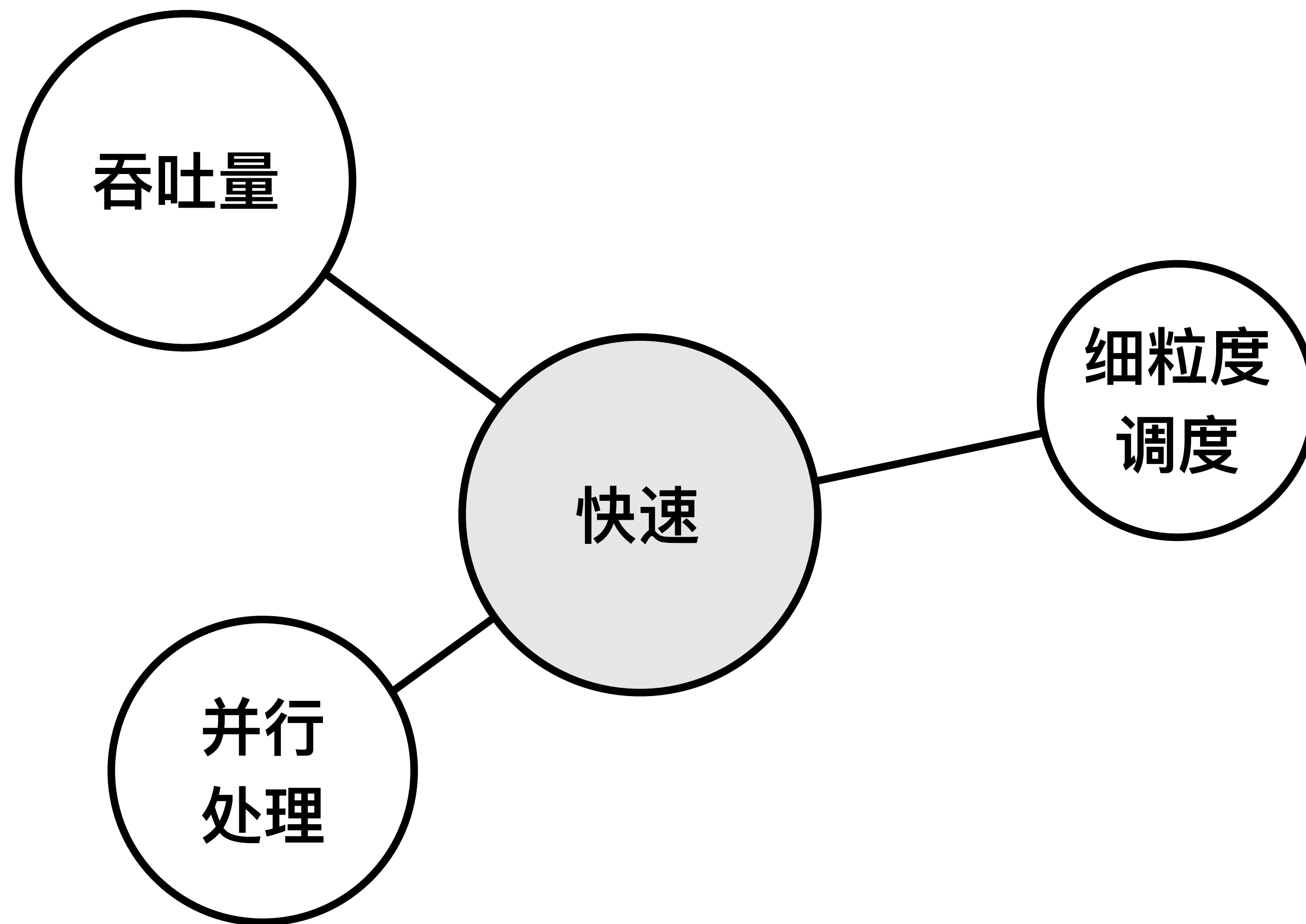
100%

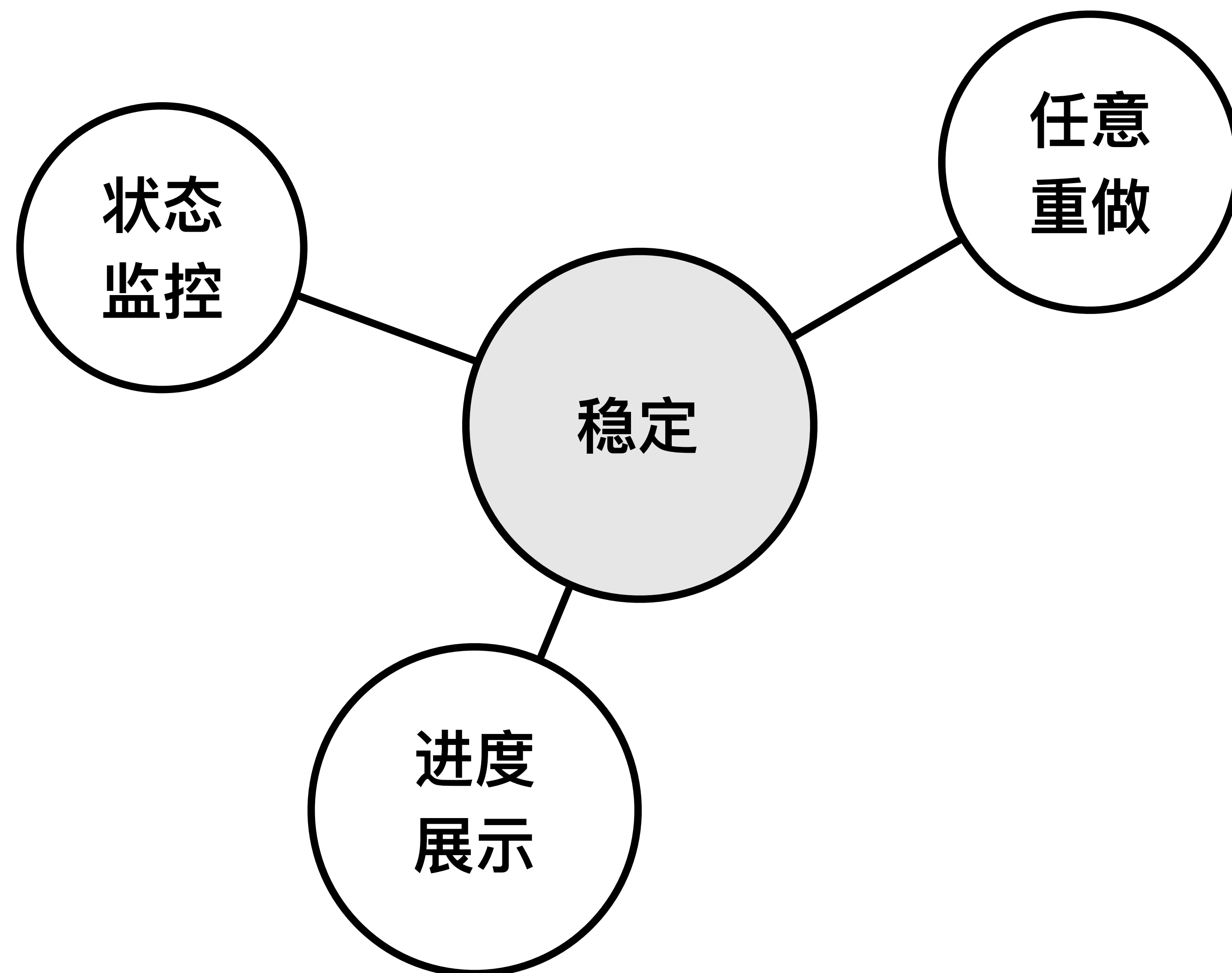


# 总结

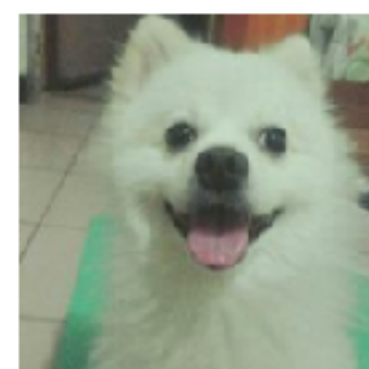








谢谢



西韭

扫描上面的二维码，关注我吧