

# 状态机的通用设计和实现

郭宁

guoning.gn@gmail.com

2017/07/07

# 目标

关于状态机的设计和实现：

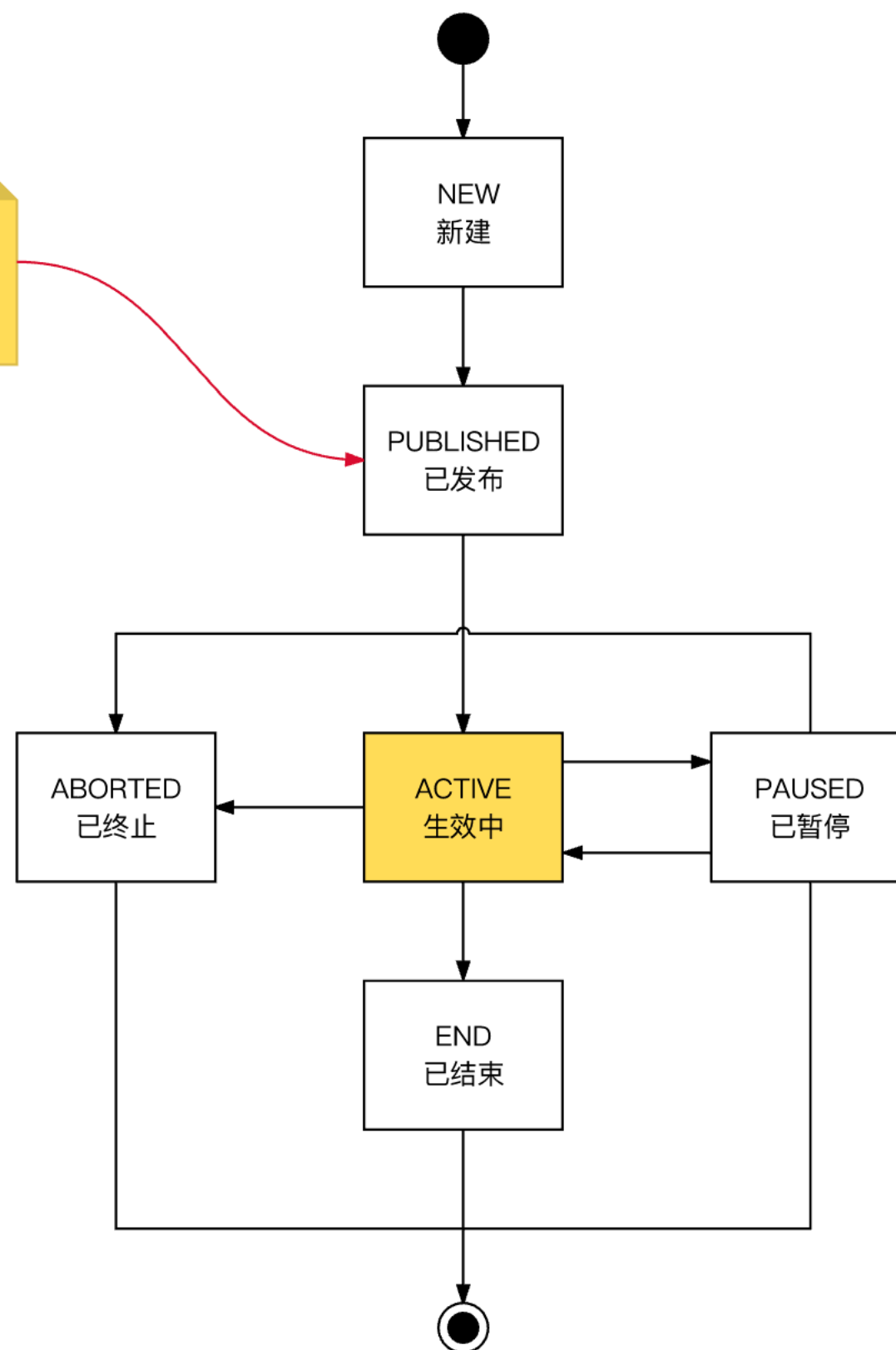
1. 基本原则
2. 可行步骤
3. 最佳实践

面向对象：

- 技术团队
- 产品团队
- 业务、技术感兴趣的同学

PM 暂时策略：

1. 「新建」状态以及之后状态，不允许编辑
2. 「已发布」状态以及之前状态，可以删除





# 状态机的通用设计和实现

郭宁

guoning@mobike.com

2017/07/07

# 目录

- 简介
- 实例：实验平台
  - 场景
  - 分析
  - 设计
  - 实现
- 小结

# 状态机设计和实现：简介

基本原则：

1. 状态互斥
2. 严格流转

关键点：

1. 状态：
  - a. 暂态
  - b. 终态
2. 触发方式：
  - a. 时间触发
  - b. 事件触发

# 状态机设计和实现：简介

状态机设计的可行步骤：

1. 分析：业务场景抽象，收集状态变更的条件
2. 设计：状态流转
3. 实现

Note：

1. 软件设计原则：SRP



# 实例：实验平台-场景

## 场景

实验平台中，实验的状态流转。

## 目标：

1. 怎么设计一个状态机，平滑的支持业务？
2. 怎么实现？实现细节上，有没有问题？

# 实例：实验平台-分析

对象实体：实验

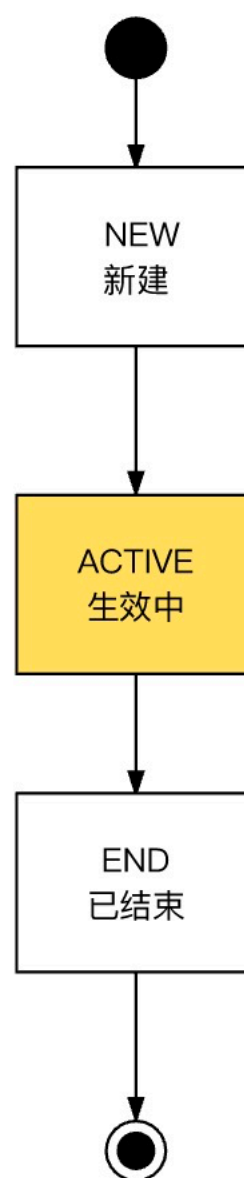
实体的状态变更条件：

1. 创建
2. 生效：开始时间，实验自动生效
3. 暂停：人为操作
4. 恢复：人为操作
5. 结束：结束实践，实验自动结束
6. 上线：人为操作
7. 删除：人为操作

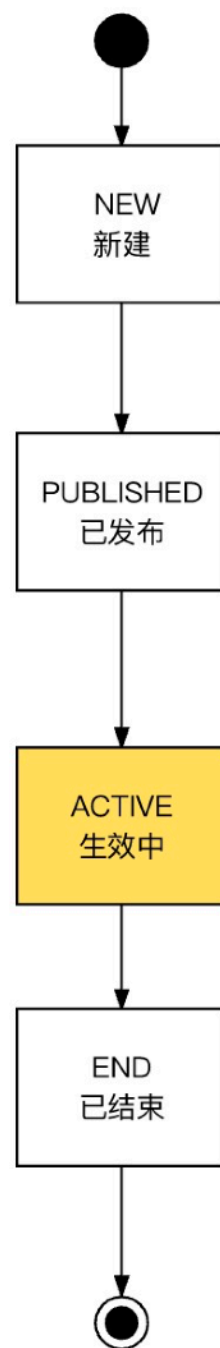
实体，不同状态下的权限：

1. 是否可编辑
2. 是否可删除
3. etc.

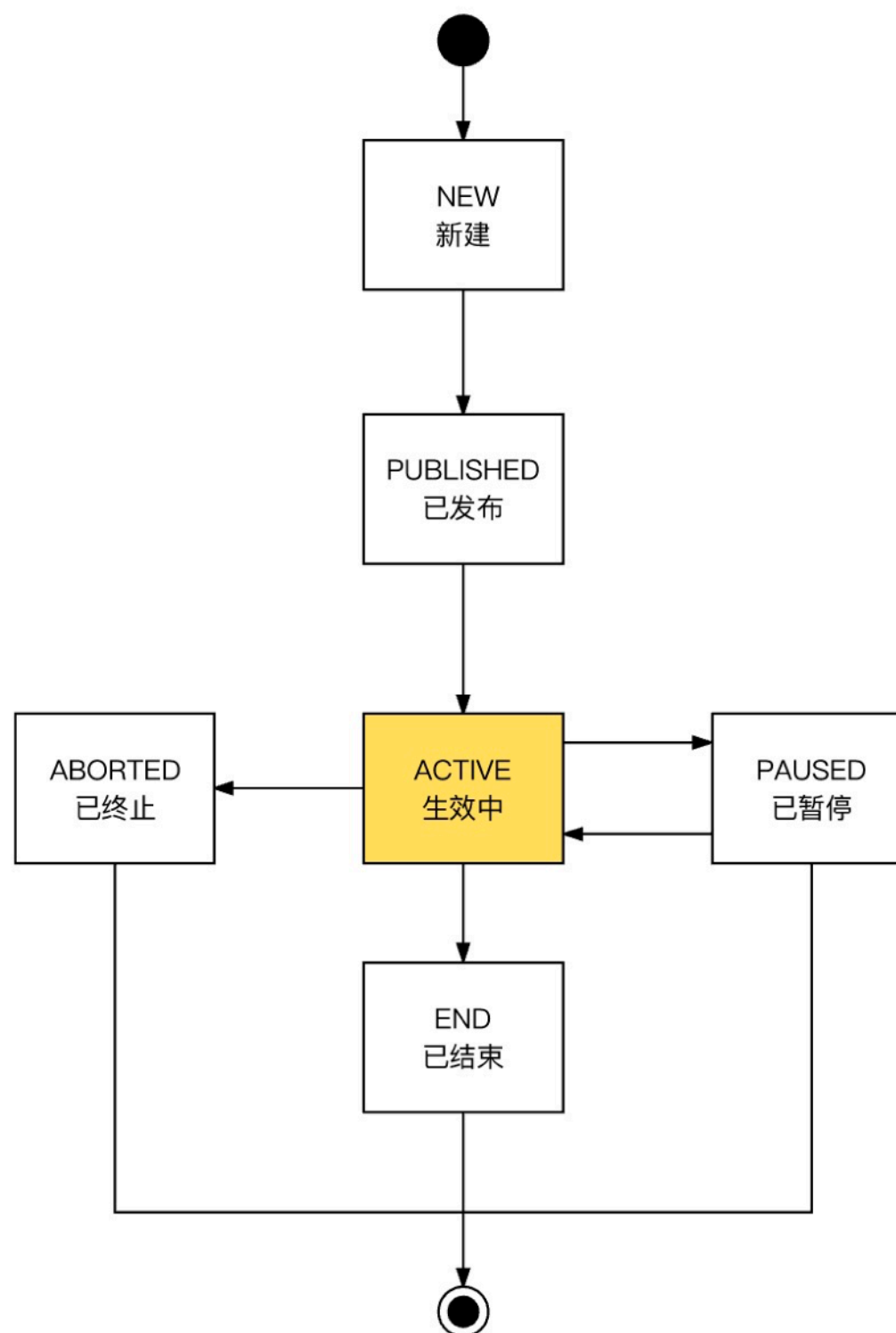
# 实例：实验平台-设计



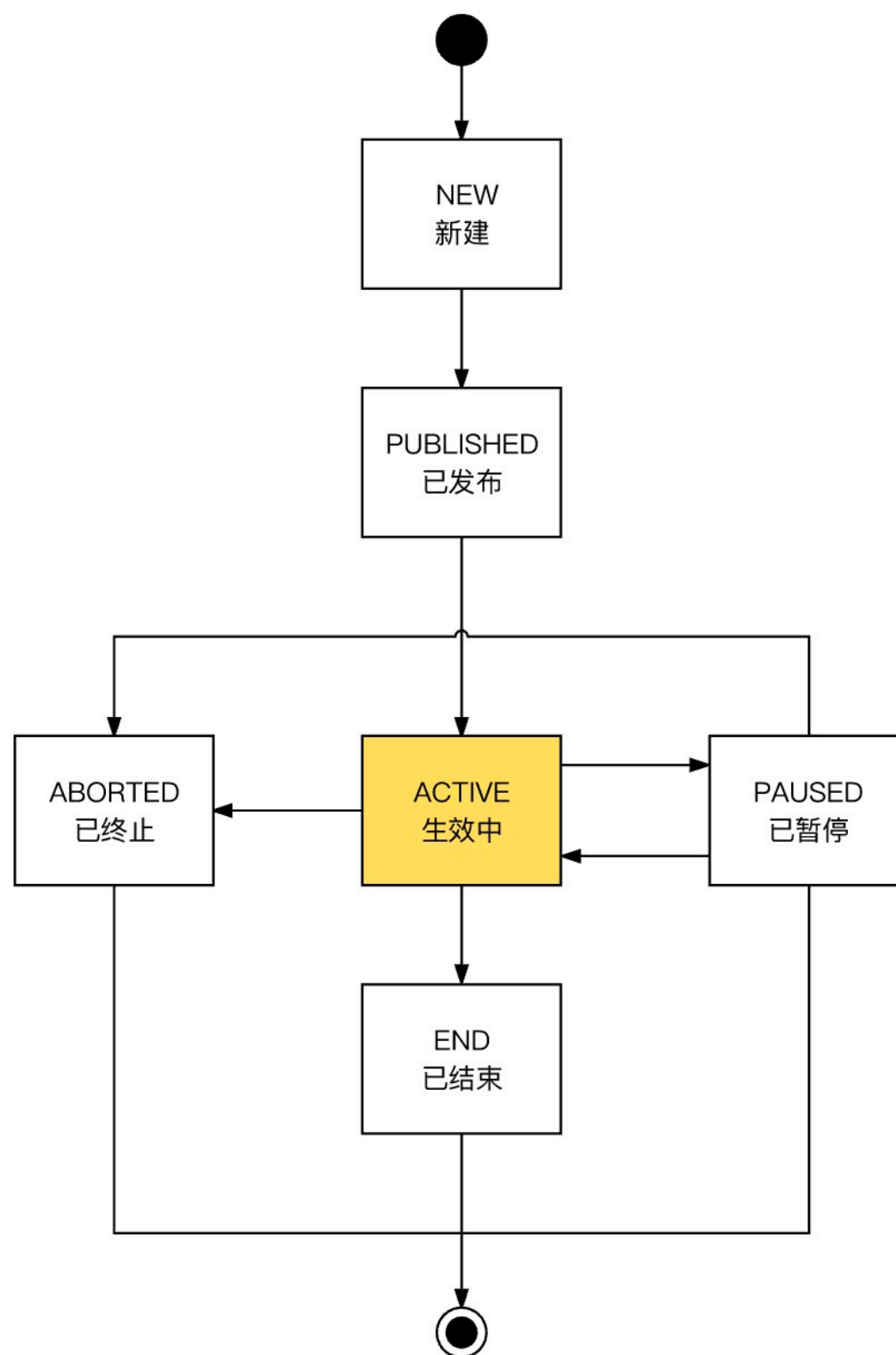
# 实例：实验平台-设计



# 实例：实验平台-设计



# 实例：实验平台-设计



# 实例：实验平台-实现

软件设计原则：

- **SRP** (Single Responsibility Principle)

状态流转

- 触发方式：
  - **时间触发**：自动流转
  - **事件触发**：人为操作

实现的部分细节：

1. 状态是否实时变更

# 小结

基本原则：

1. 状态互斥
2. 严格流转

关键点：

1. 状态：暂态、终态
2. 触发方式：时间触发、事件触发

状态机设计的可行步骤：

1. 分析：业务场景抽象，收集对象状态变更的条件，
  - a. 对象：是什么？
  - b. 状态变更条件：什么情况下，状态会变更？
  - c. 权限：不同状态下，可以进行哪些操作？
2. 设计：状态流转
3. 实现

Note：

1. 软件设计原则：SRP