

# Migrating a monolith application to microservices

# Raiffeisen OnLine

- **Internet Service Provider**
- Internet Access: ADSL, VDSL, Fiber
- VoIP: Private and Business Solutions
- Hosting: Webspace, Virtual Servers
- Software Development: Internal and External Projects
- ...



<https://raiffeisen.net>



alex.lanz@raiffeisen.net



@AlexKrumer



Raiffeisen Online

# Administration Application

- Used daily by every employee
- Internal Application
  - Manage Customers
  - Manage Sales
  - Manage Services
  - ...



<https://raiffeisen.net>



alex.lanz@raiffeisen.net

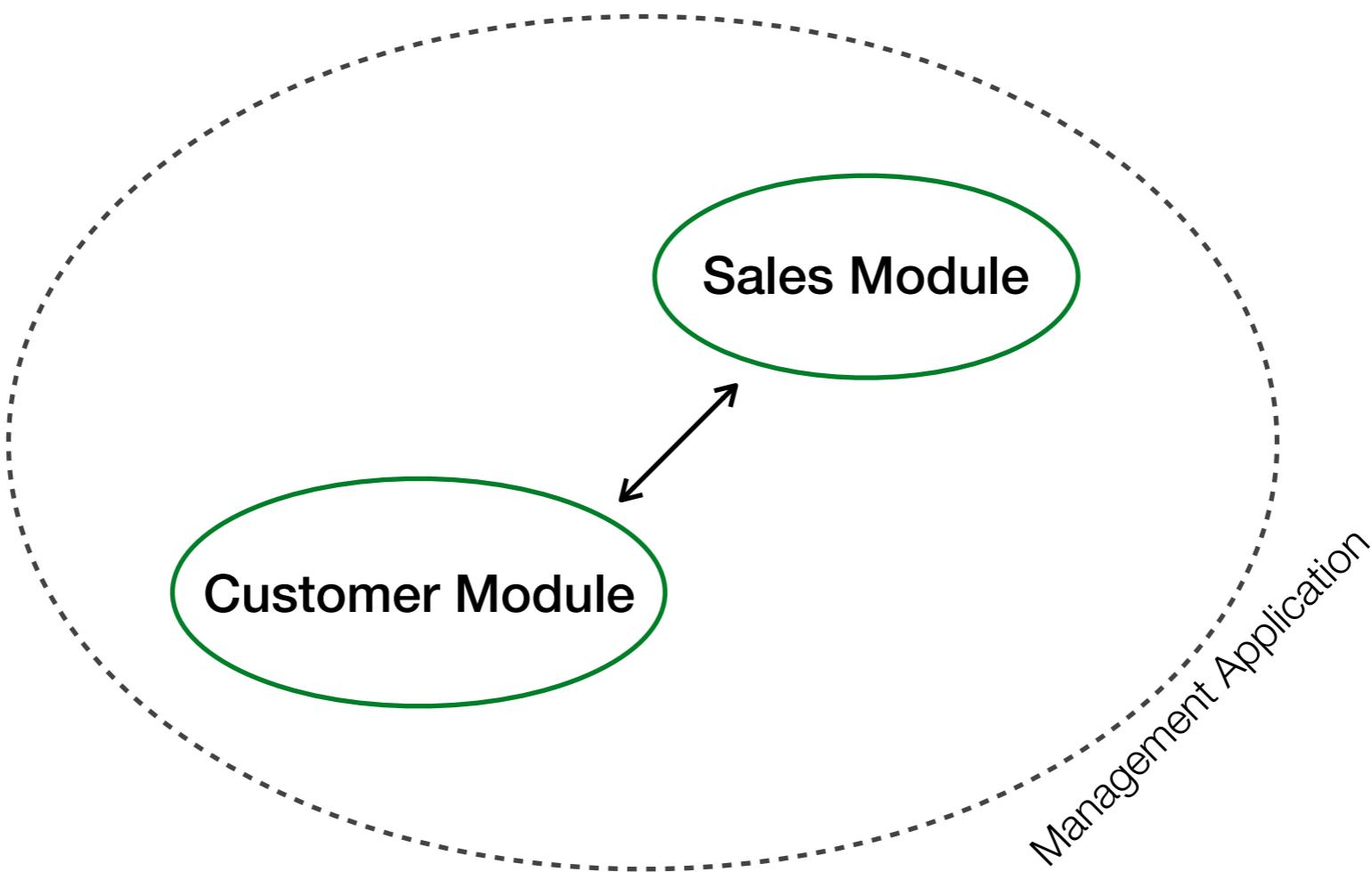


@AlexKrumer

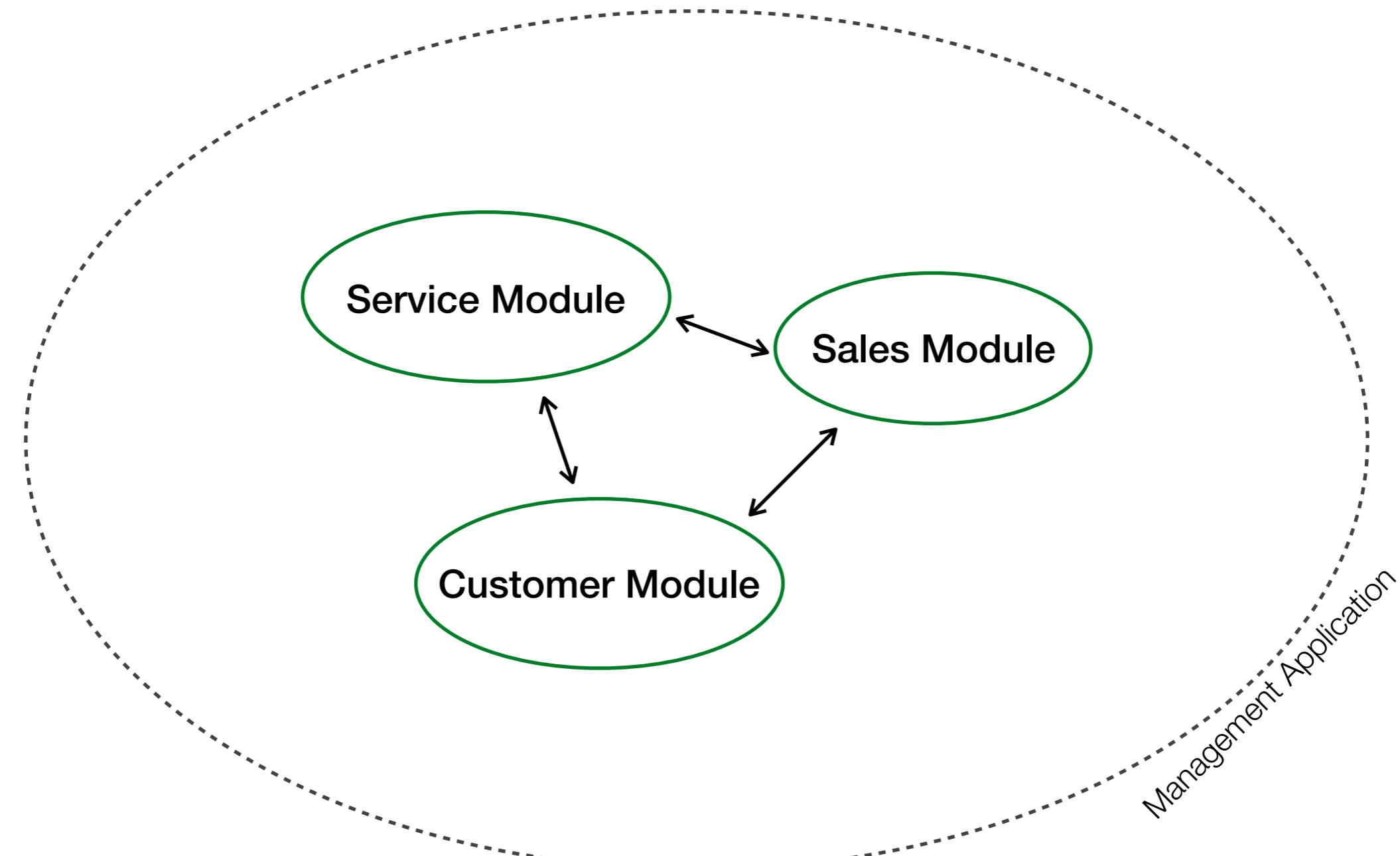


RaiffeisenOnline

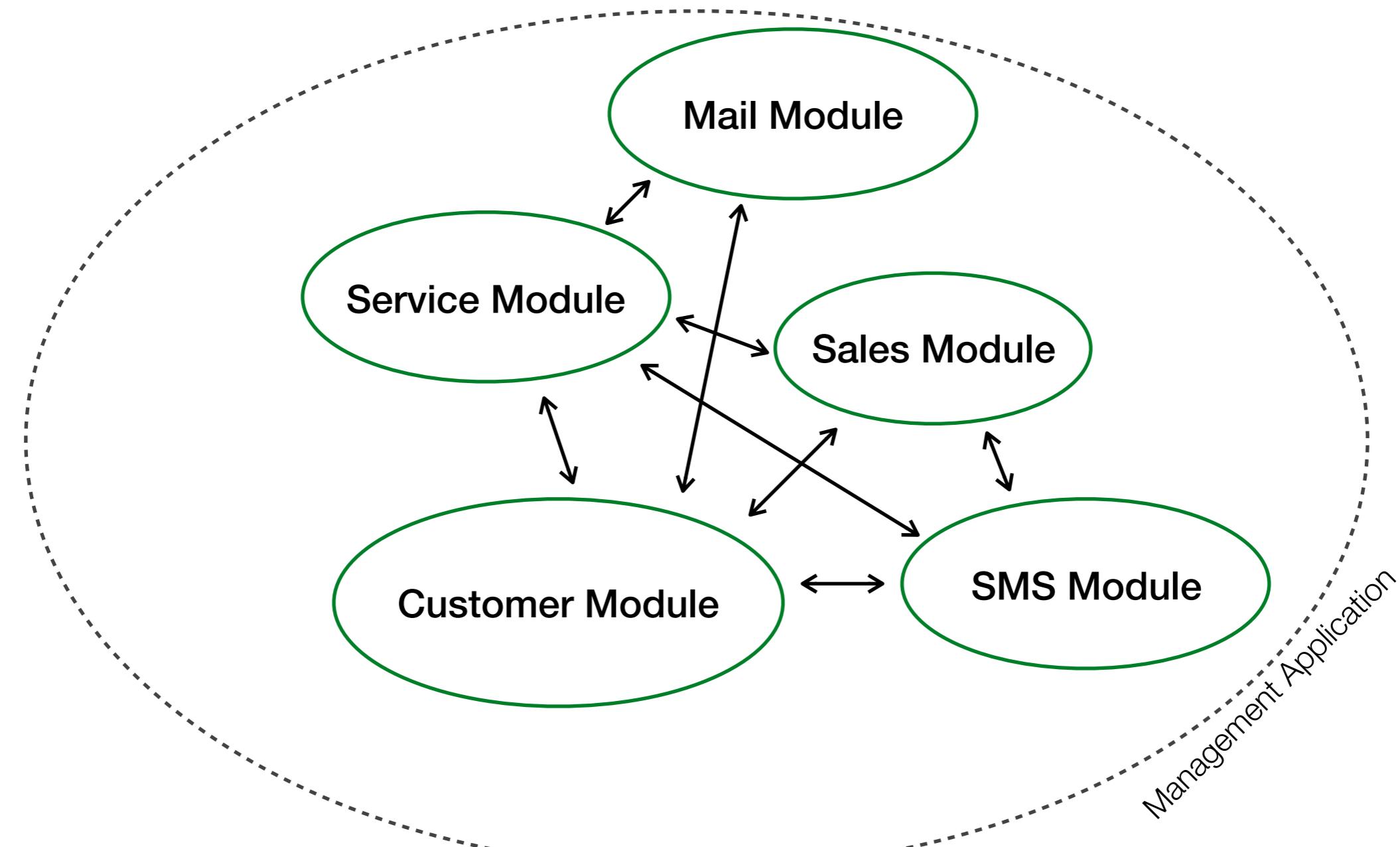
# Over the Years...



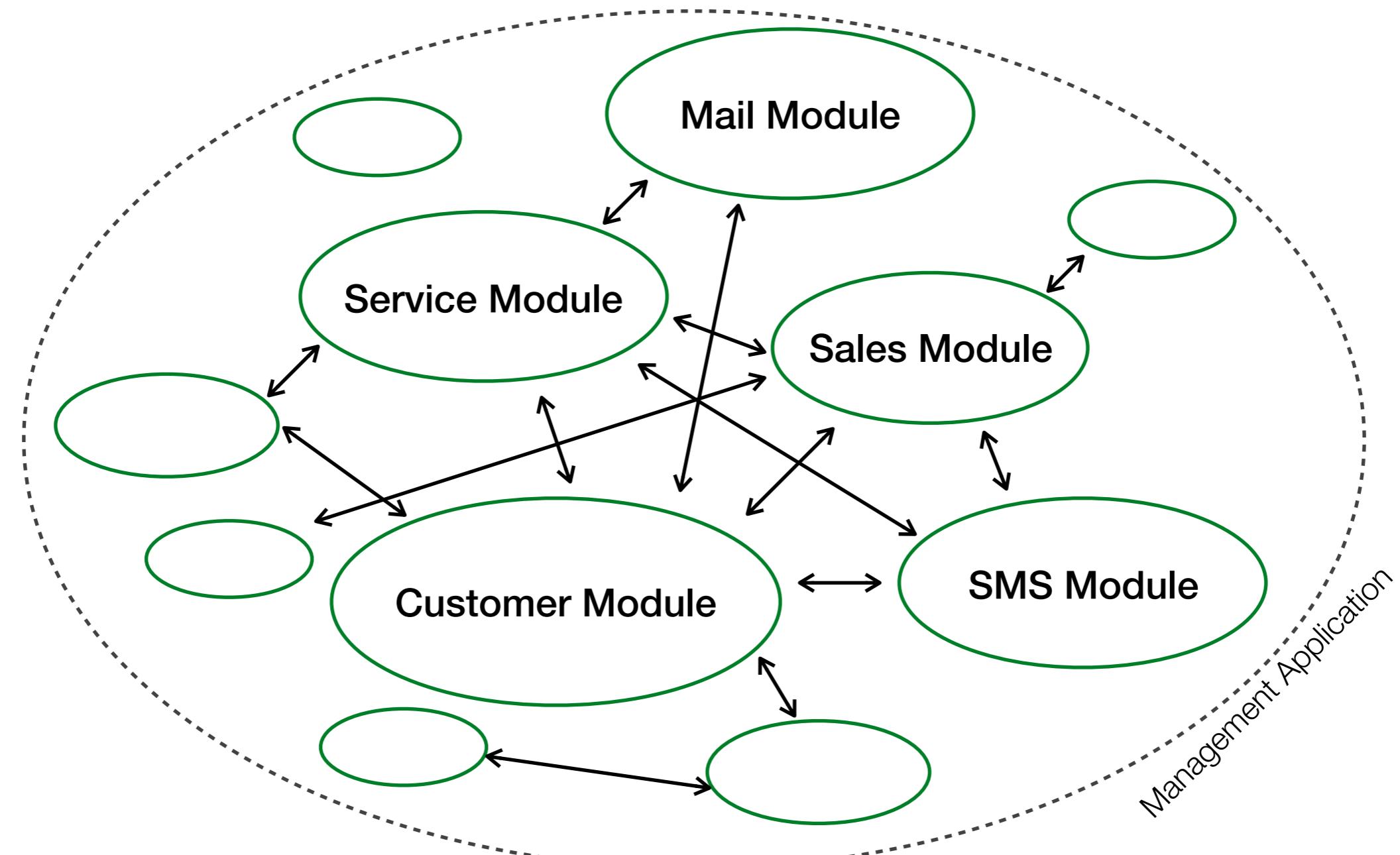
# Over the Years...



# Over the Years...



# Over the Years...



# Let's refactor everything

- Several attempts were made in the past
- Refactoring the whole application
- Rewrite from scratch



<https://raiffeisen.net>



alex.lanz@raiffeisen.net



@AlexKrumer



Raiffeisen Online

# Let's refactor everything

- Several attempts were made in the past
- Refactoring the whole application
- Rewrite from scratch

➡ All attempts failed

# Let's refactor everything

- Several attempts were made in the past
- Refactoring the whole application
- Rewrite from scratch

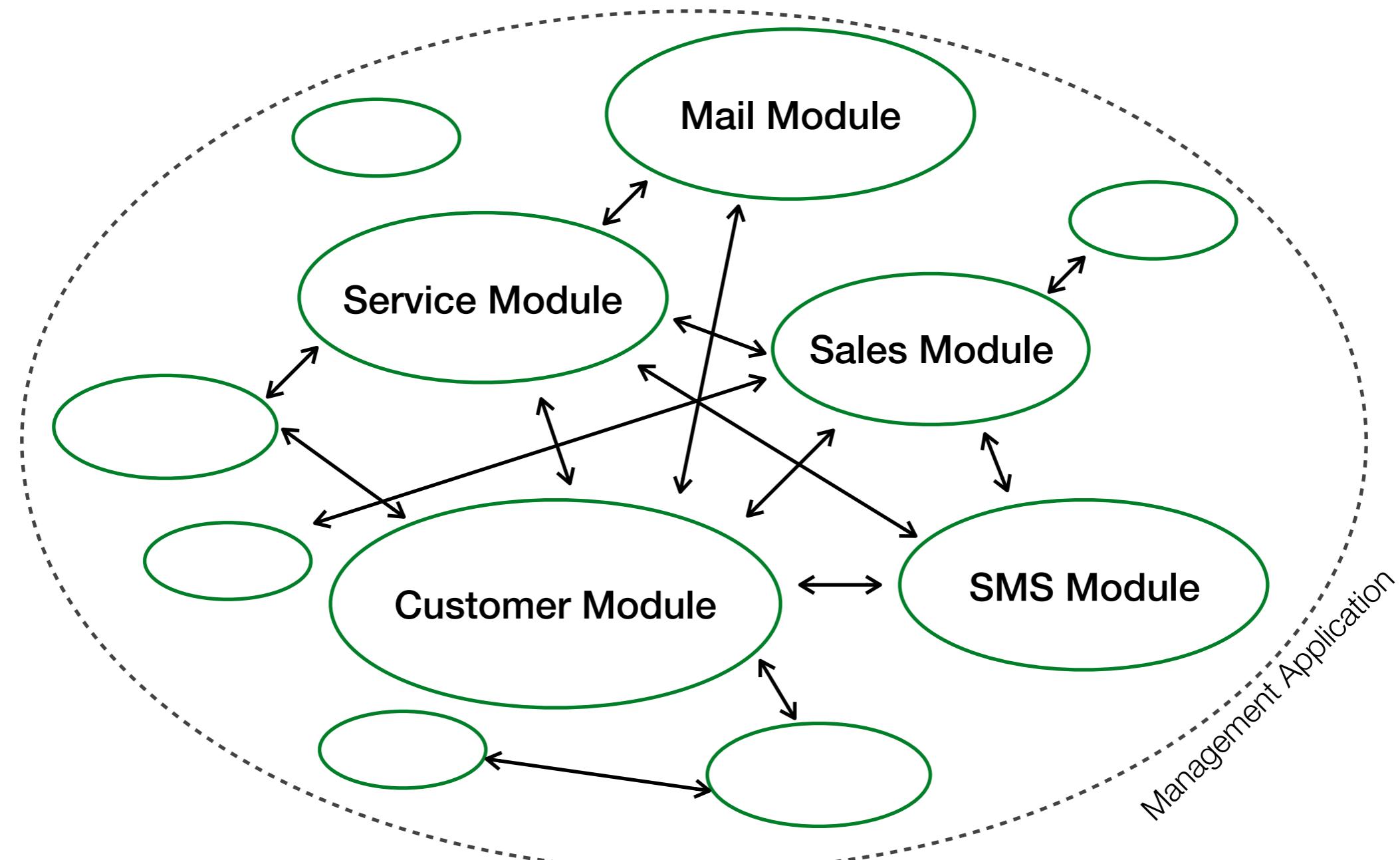
➡ All attempts failed

➡ They made the code even worse

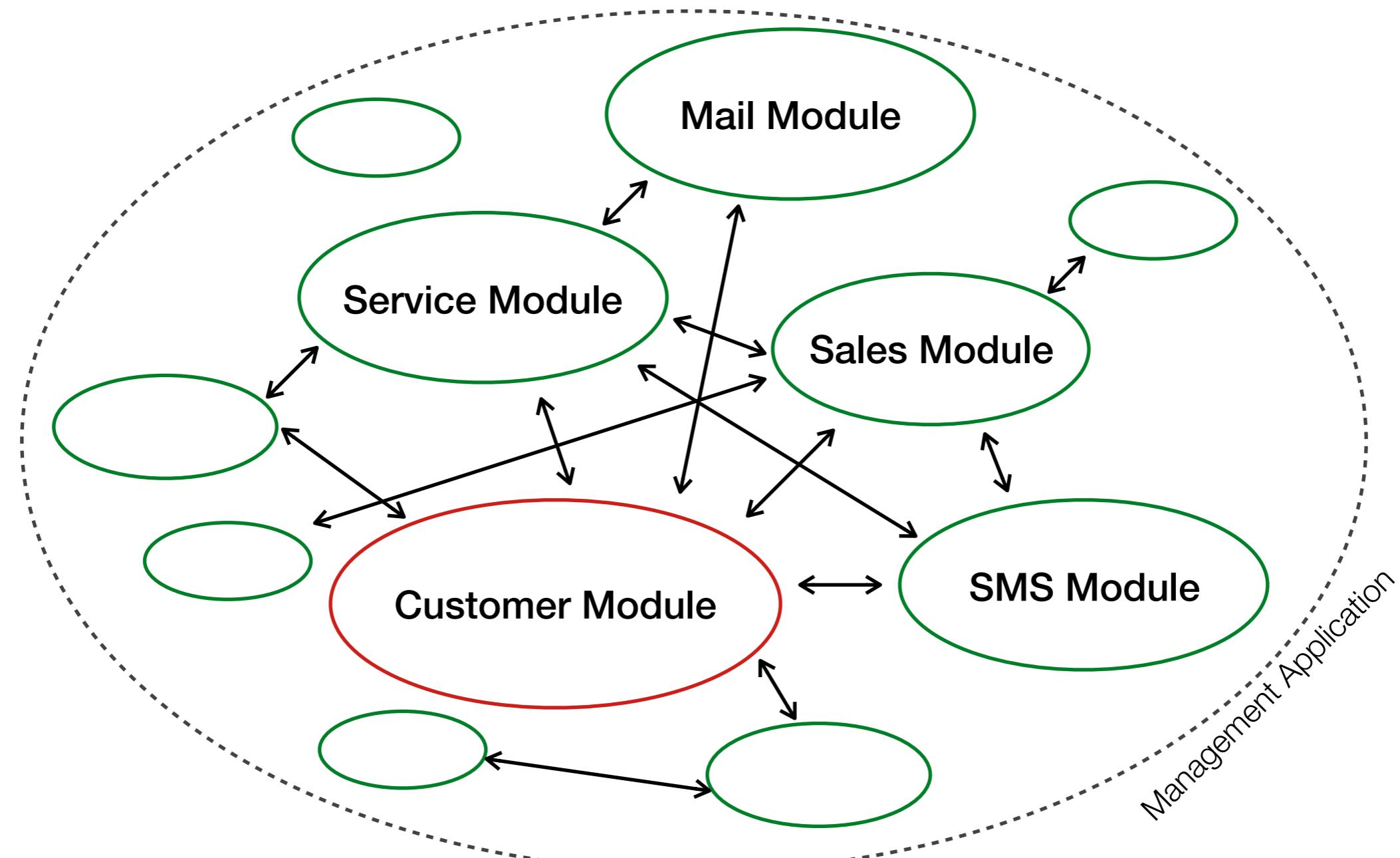
# But how can we update?

1. Analyze and group parts of the application
2. Try to identify boundaries between the parts
  - Group based on high cohesion
  - Group based on loose coupling
3. Extract on part at the time

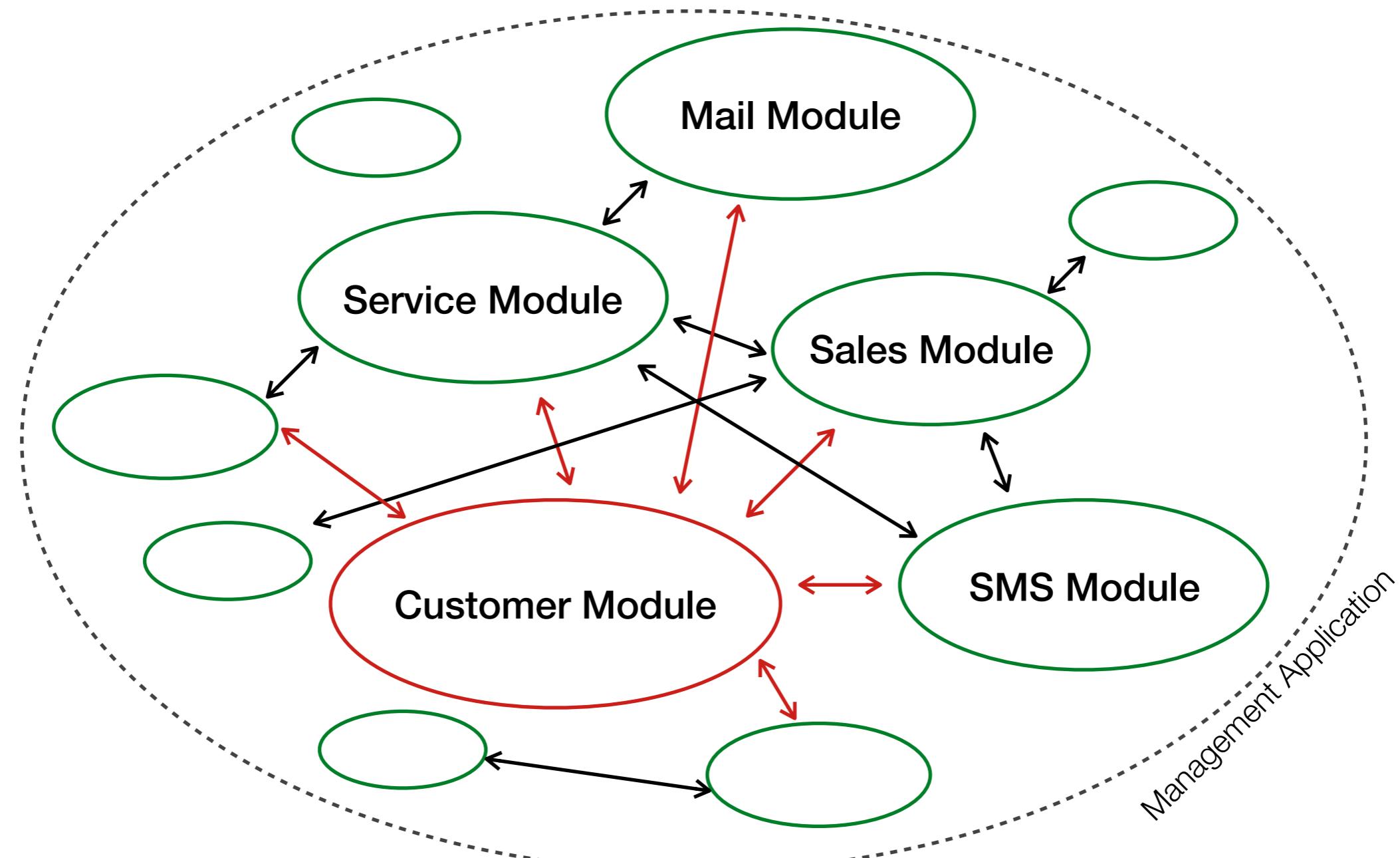
# But how can we update?



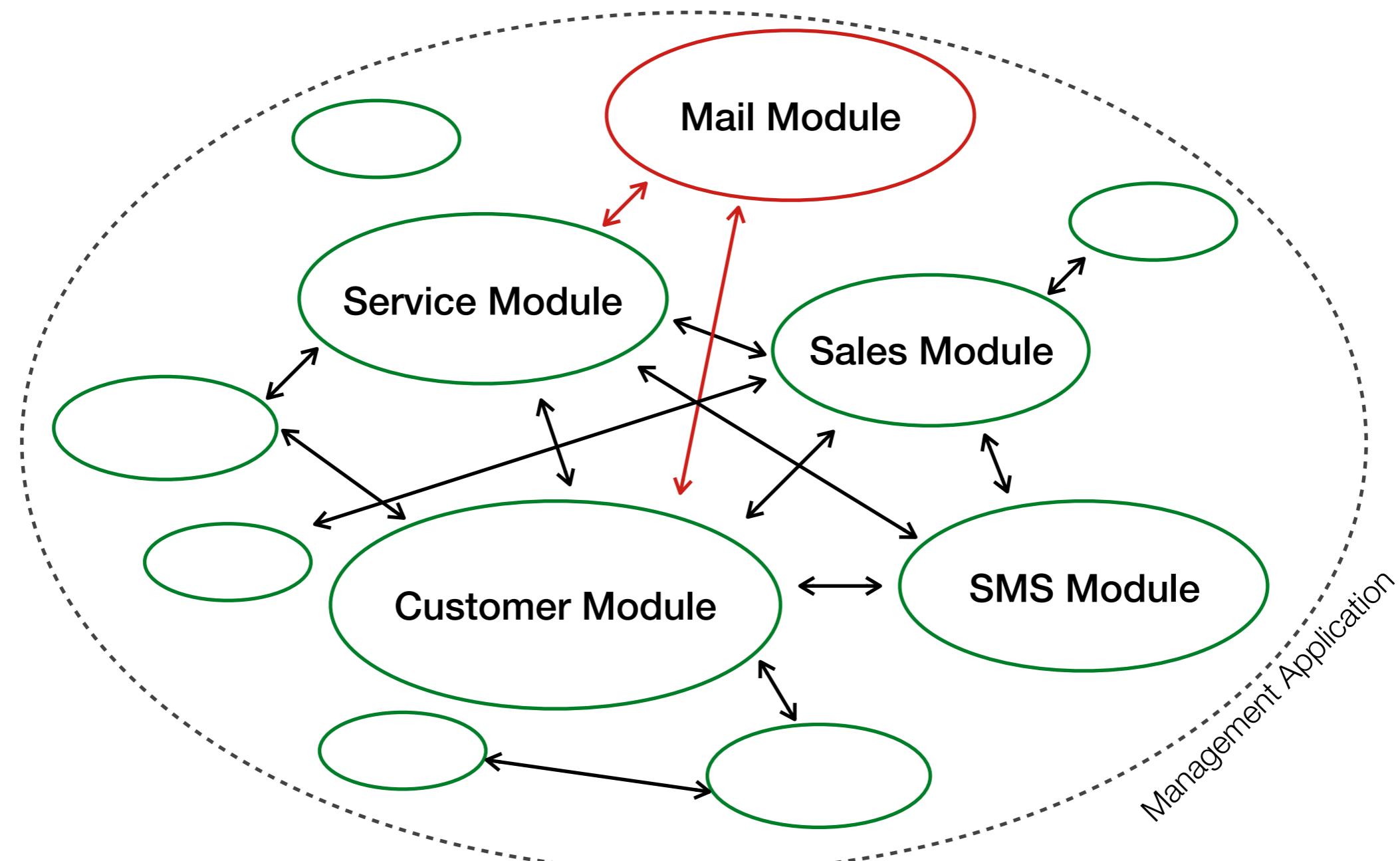
# But how can we update?



# But how can we update?



# But how can we update?



# Preparations

- Define some standards
  - Domains and certificates
  - Authentication (Basic Authentication with SSL)
  - Request and response structure (REST, JSON)



<https://raiffeisen.net>



alex.lanz@raiffeisen.net



@AlexKrumer



RaiffeisenOnline

# Preparations

- Create some packages
  - Logging
  - API server (server package for creating API responses)
  - API client (client package for requesting APIs and process their responses)



<https://raiffeisen.net>



alex.lanz@raiffeisen.net



@AlexKrumer



Raiffeisen Online

# Preparations

- Create a boilerplate for new microservices
  - Development Environment
  - Tests
  - Logging
  - Monitoring
  - Deployment



<https://raiffeisen.net>



alex.lanz@raiffeisen.net



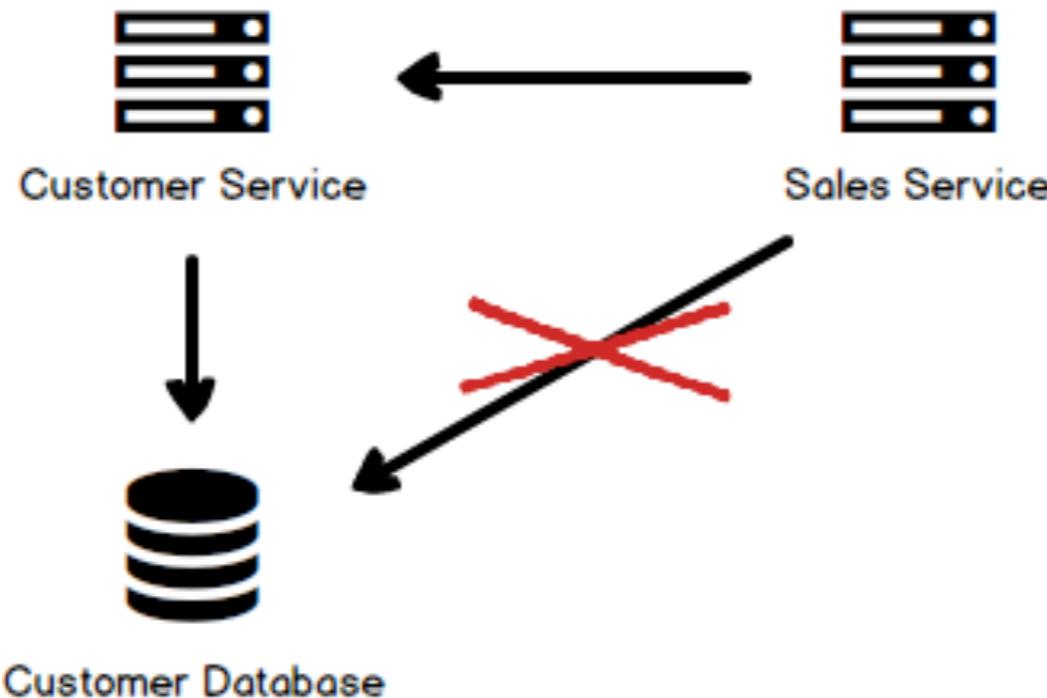
@AlexKrumer



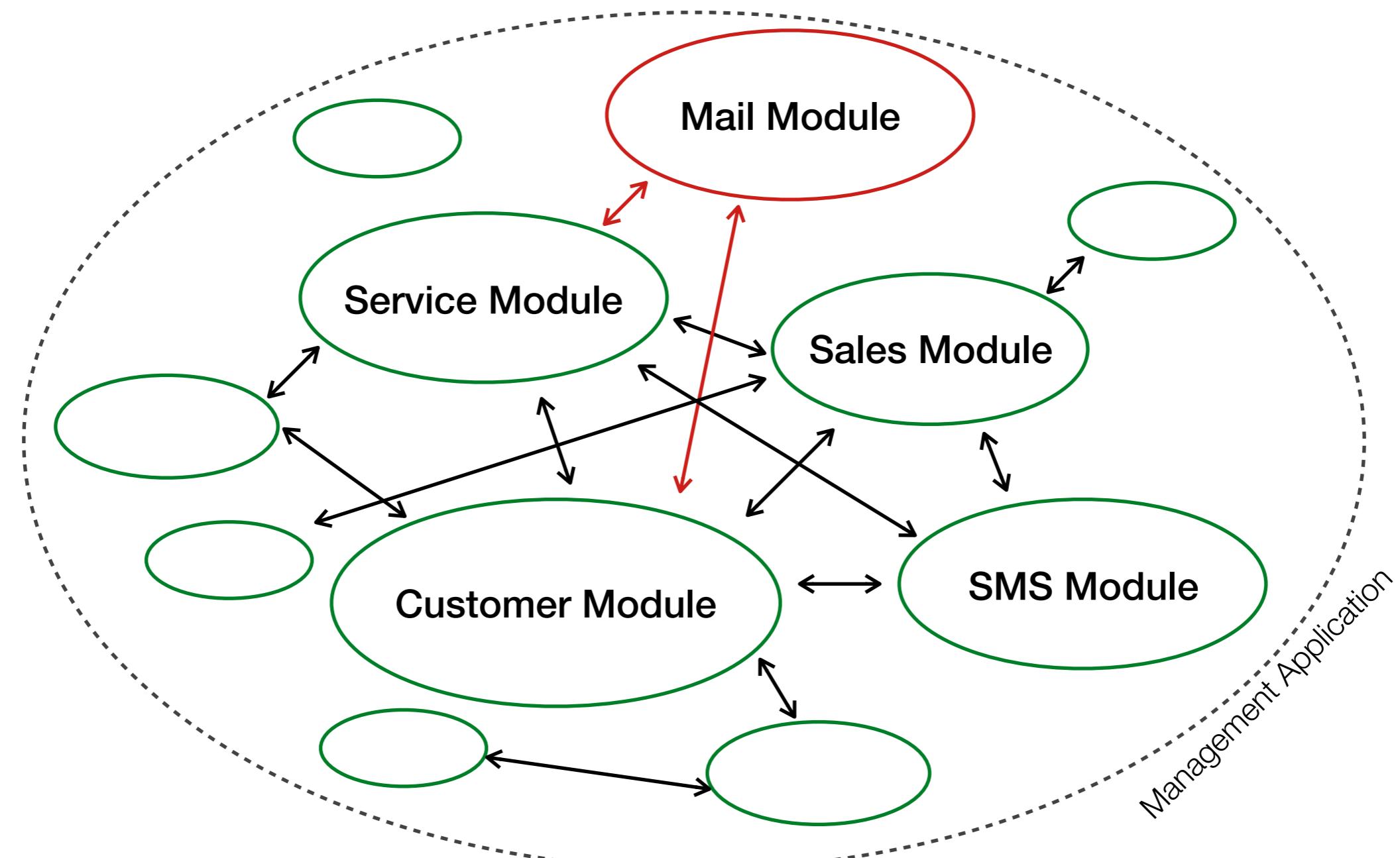
Raiffeisen Online

# Rules

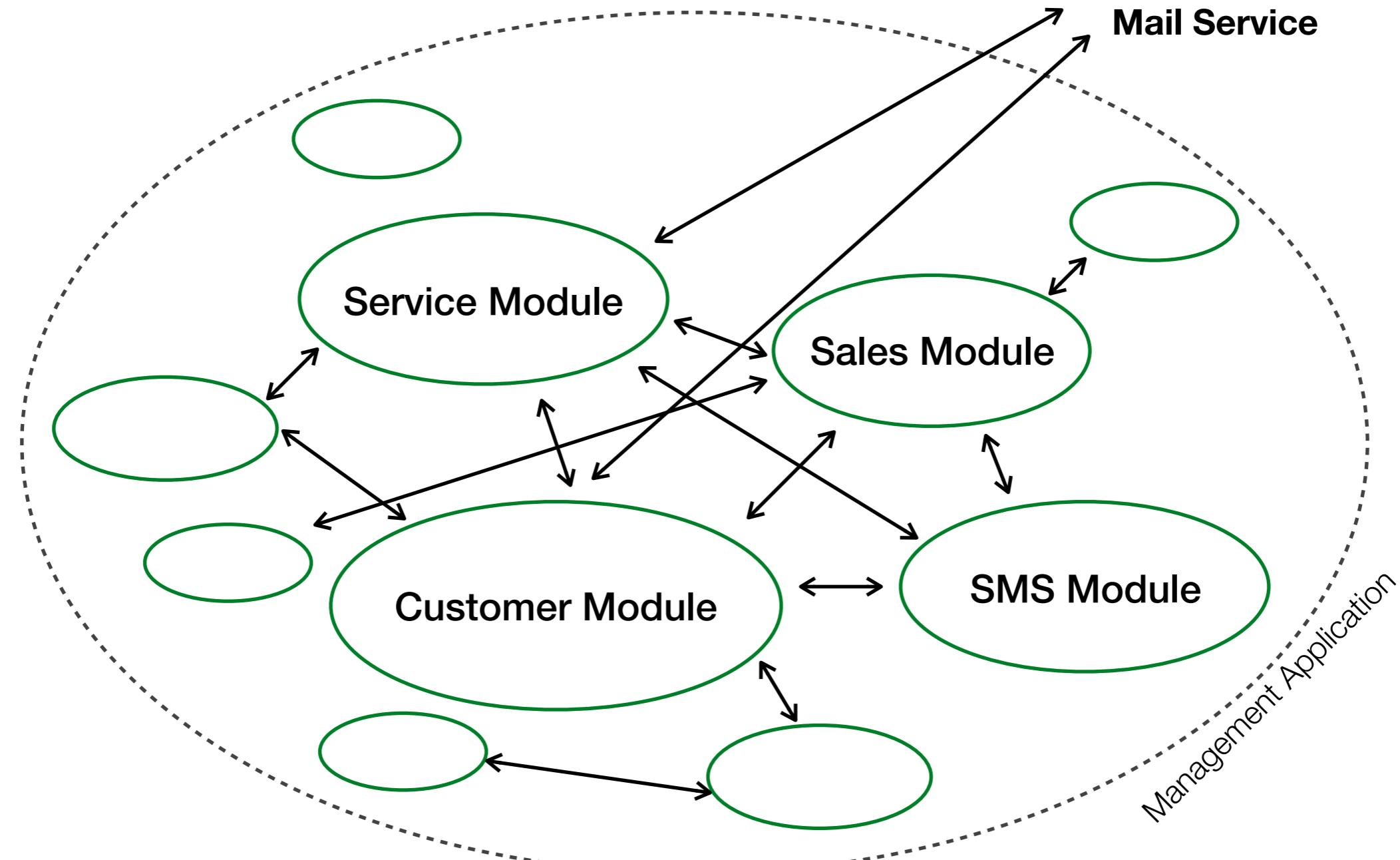
- Only the responsible service can access the database
- Other services have to access the API



# Extraction



# Extraction



# Our Microservices

## Already migrated:

- Country Service
- Mail Service
- Domain Service
- Product Service
- Fax Service

## Currently under development:

- Payment Service
- Customer Service

## Planned to be migrated:

- Sales Service
- ...



<https://raiffeisen.net>



alex.lanz@raiffeisen.net



@AlexKrumer



Raiffeisen Online

# Open Challenges

- Continuous Documentation
  - How to keep the overview?
- ...



<https://raiffeisen.net>



alex.lanz@raiffeisen.net



@AlexKrumer



RaiffeisenOnline

# Takeaways

- Setup a good foundation (boilerplate, standards, rules)
- Don't violate your defined rules
- Identify parts with high cohesion and loose coupling
- Split up the monolith application step by step



<https://raiffeisen.net>



alex.lanz@raiffeisen.net



@AlexKrumer



Raiffeisen Online