ACL в ERP/CRM системах: архитектура и реализация на конкретном примере

Кузовлев Евгений

Проектирование ACL - вопросы разработчика

- Модели систем управления доступом – какую выбрать?
- Архитектура приложения – как интегрировать ACL в систему?
- Что должно быть объектом доступа в вашей системе?
- Как реализовать все требования заказчика и при этом сделать систему управляемой и расширяемой?

Модели контроля доступа

- Черный ящик — ACL::checkAccess()
- Возможные модели:
  - Discretionary access control (DAC)
  - Mandatory access control (MAC)
  - Role-based access control (RBAC)
  - Attribute-based access control (ABAC)

DAC
Избирательное управление доступом

- Пользователи (субъекты)
- Набор действий (объектов), к которым регламентируется доступ
- Таблица доступа (матрица, список), описывает каждую пару субъект-объект

Пример: Atlassian Wiki

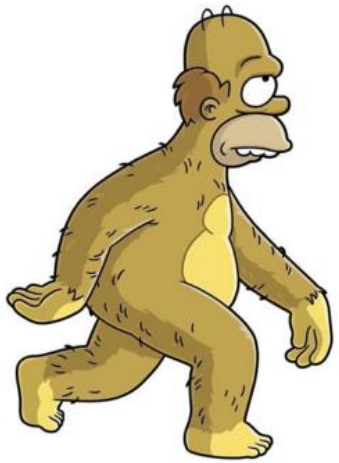
MAC
Мандатное управление доступом

- Пользователи (субъекты), у каждого есть роль
- Набор действий (объектов)
- Уровень доступа субъекта
- Уровень доступа объекта
- Иерархия уровней
- Минимальный уровень для разрешения

Пример: ОС МСВС

RBAC
Управление доступом на основе ролей

- Пользователи (субъекты)
- Набор действий (объектов), к которым регламентируется доступ
- Набор ролей
- Таблица доступа (матрица, список), описывает каждую пару роль-объект



Пример: ACL в феймворке Yii

RBAC + MAC
Многоуровневое управление доступом

- Пользователи (субъекты)
- Набор действий (объектов), к которым регламентируется доступ
- Набор ролей
- Таблица доступа (матрица, список), описывает уровень доступа для каждой пары роль-объект

ABAC
Управление доступом на основе аттрибутов

- Пользователи (субъекты)
- Набор действий (объектов), к которым регламентируется доступ
- Набор атрибутов пользователя (роли)
- Набор правил, описывающий разрешения к объектам по атрибутам

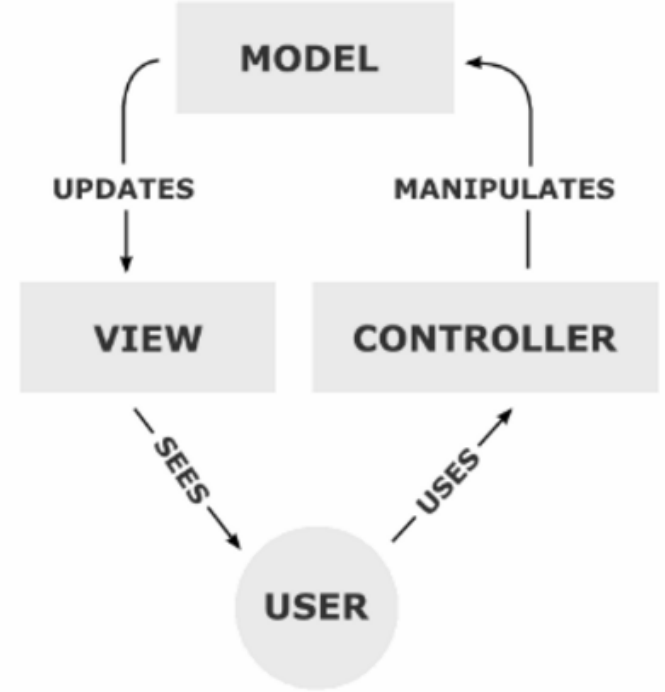
Архитектура приложения и ACL - стандартные подходы

- Прямой контроль доступа – прямые проверки там, где необходимо
- Абстракция ACL на уровне подсистемы роутинга – универсальный контроль доступа к действиям
- Абстракция ACL на уровне подсистемы доступа к данным (ORM) – универсальный контроль доступа к данным

ACL и MVC

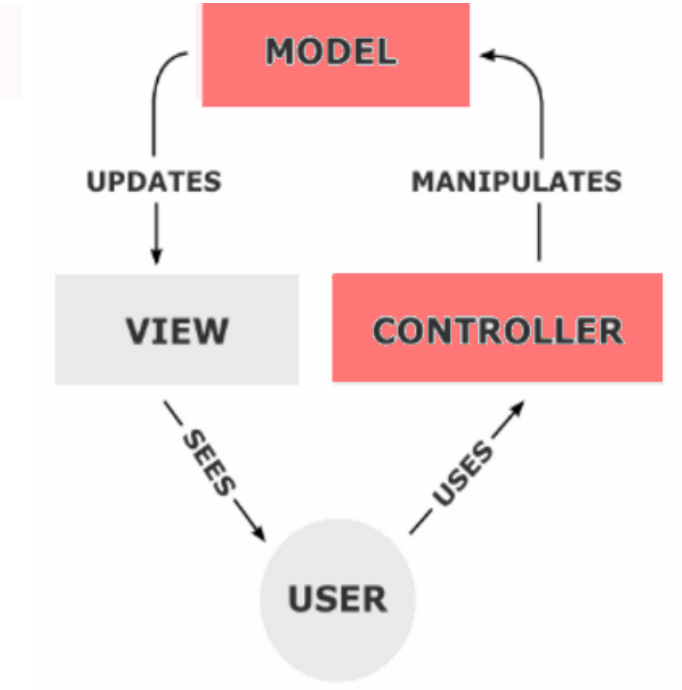
2 измерения объектов контроля:

- Деятельность: какие действия разрешены субъекту
- Доступность: какие данные в системе ему доступны



ACL и MVC: стандартный подход

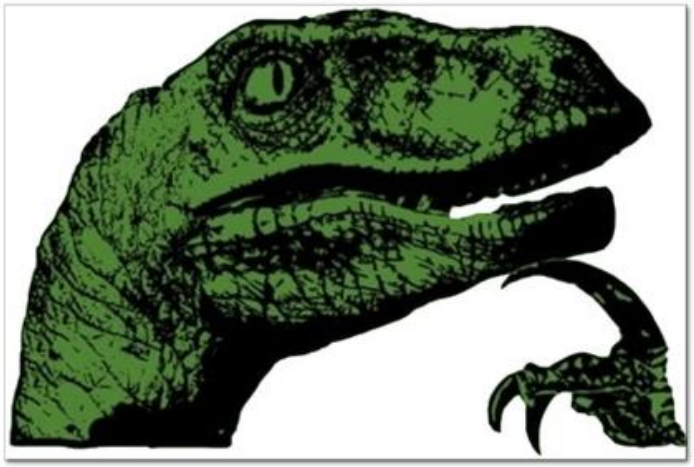
- Controller – объект контроля действий
- Model — объект контроля данных

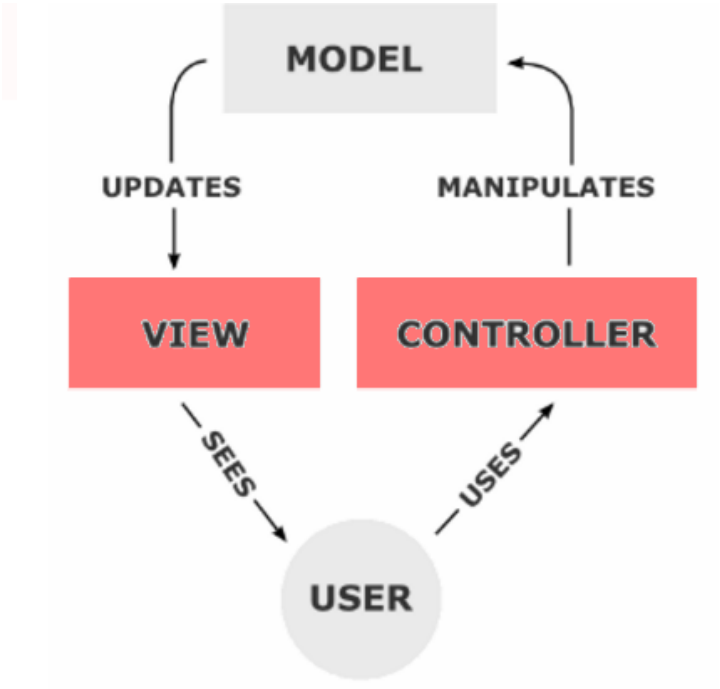


Проблемы:

- Отсутствие абстракции (в шаблонах вынуждены прибегать к прямым проверкам)
- Большое количество ошибок на этапе эксплуатации и поддержки из-за инъекций ACL в шаблонах.

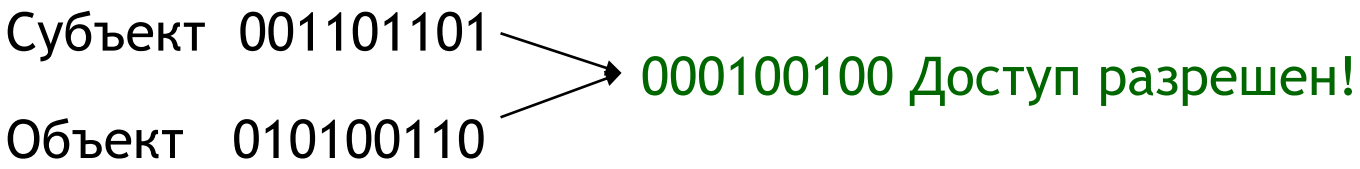
ACL и MVC: а что если?

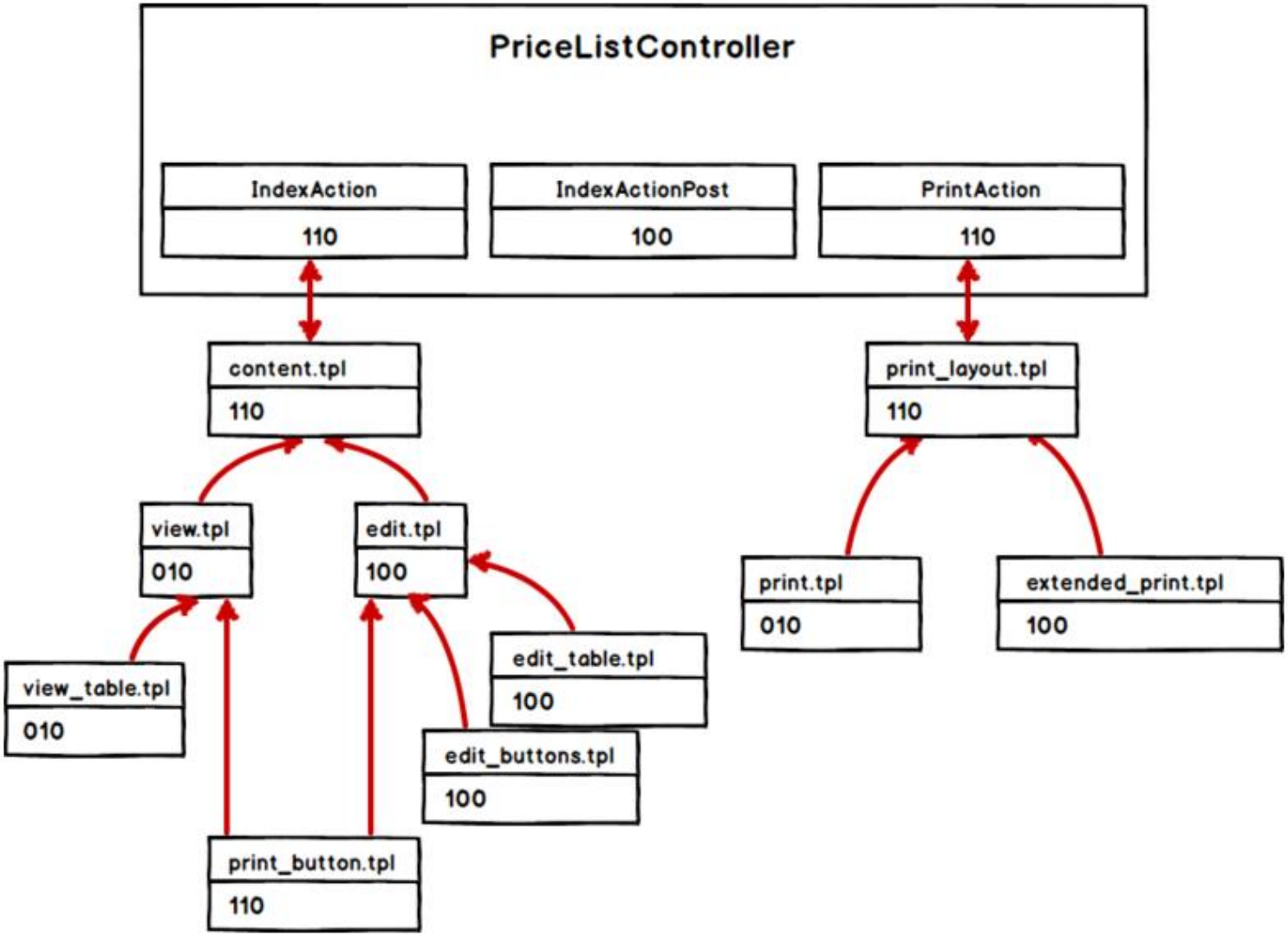


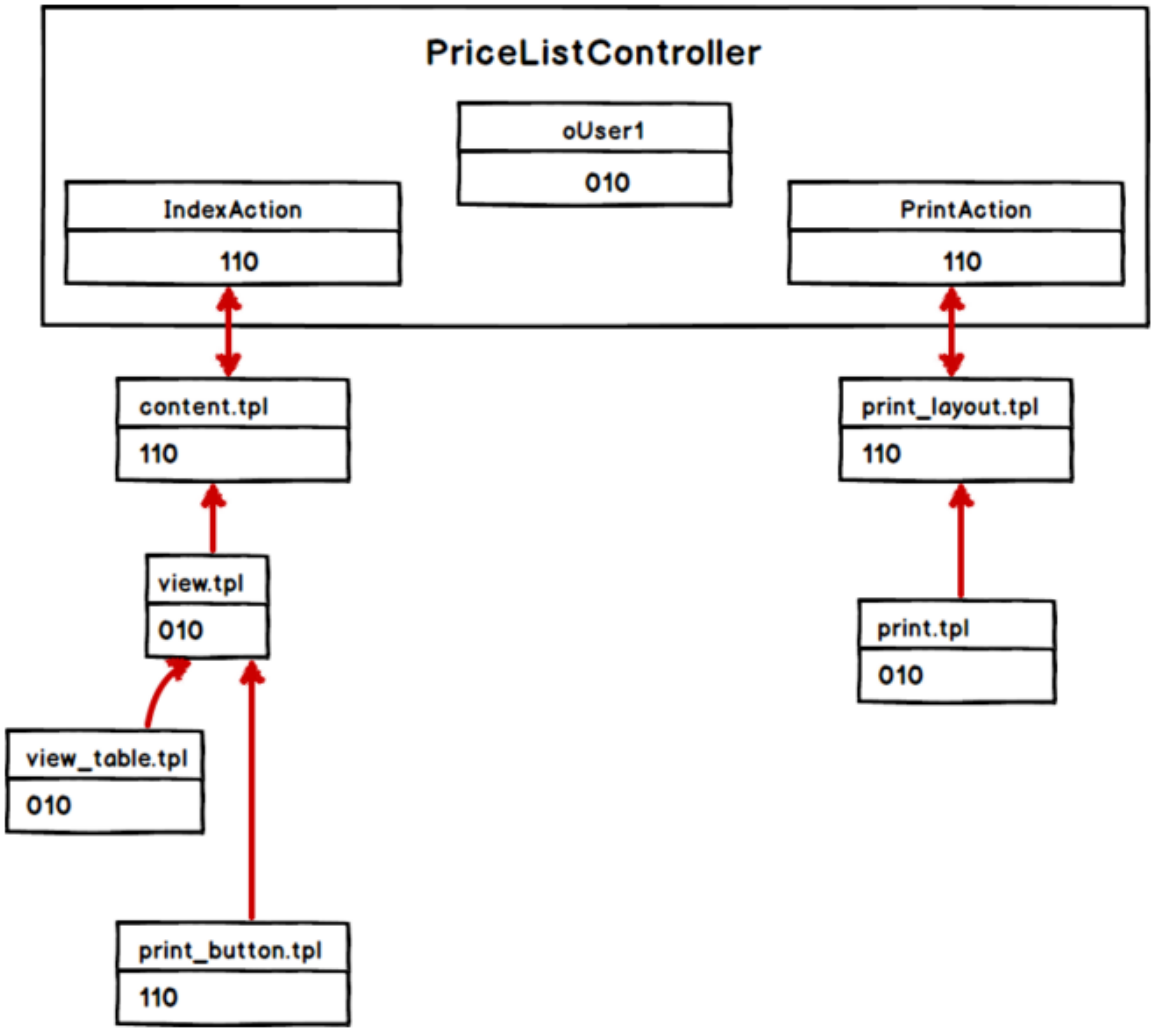


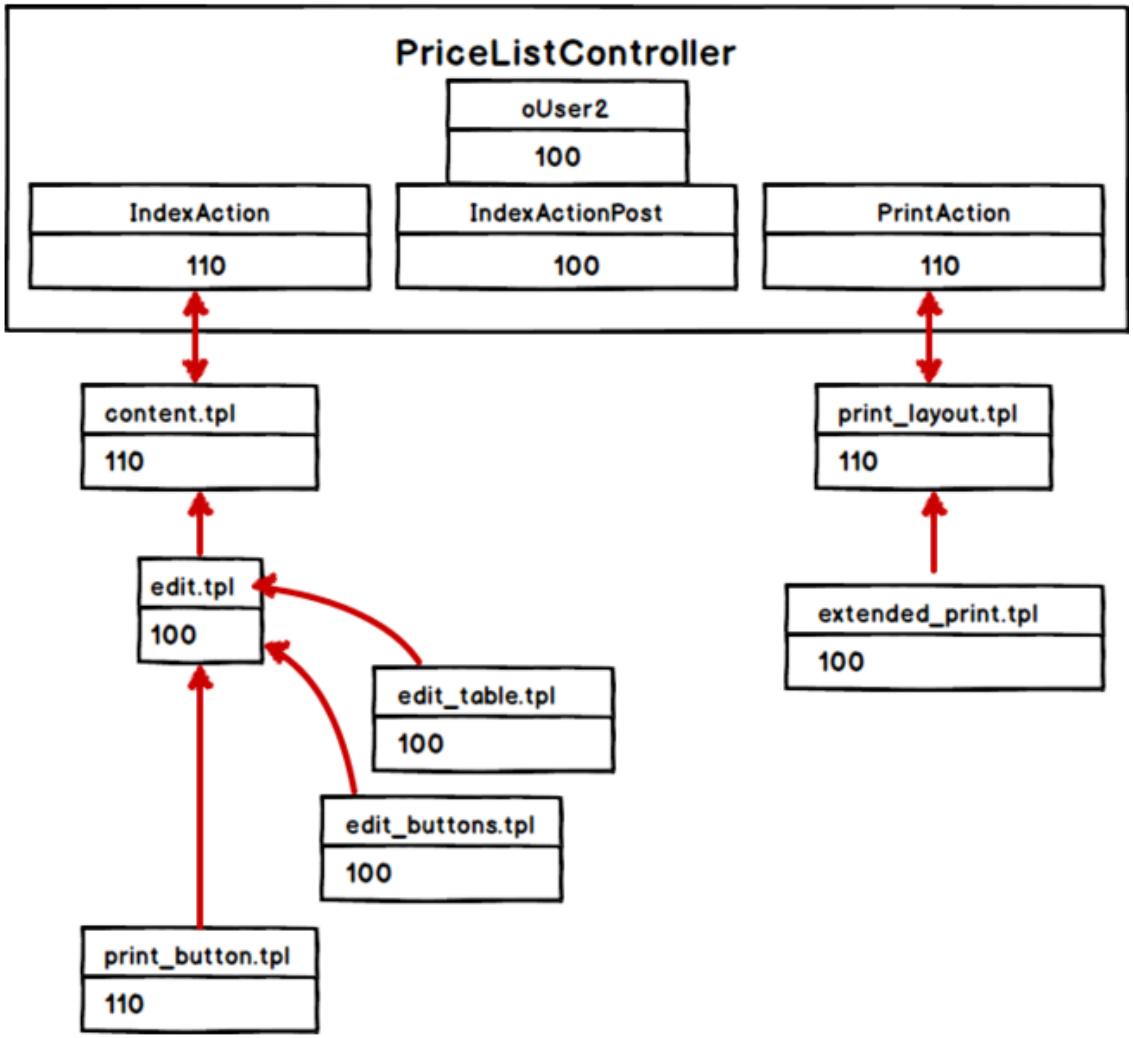
Наши решения

- RBAC модель доступа
- Бинарные вектора доступа для объектов и субъектов, один бит – одна группа доступа
- Обратные связи для обеспечения включения/выключения отдельных групп доступа для роли пользователя




PriceListController
IndexAction
110
IndexActionPost
100
PrintAction
110
content.tpl
110
print_layout.tpl
110
view.tpl
010
edit.tpl
100
print.tpl
010
extended_print.tpl
100
view_table.tpl
010
edit_table.tpl
100
edit_buttons.tpl
100
print_button.tpl
110


PriceListController
oUser1
010
IndexAction
110
PrintAction
110
content.tpl
110
print_layout.tpl
110
view.tpl
010
print.tpl
010
view_table.tpl
010
print_button.tpl
110


PriceListController
oUser2
100
IndexAction
110
IndexActionPost
100
PrintAction
110
content.tpl
110
print_layout.tpl
110
edit.tpl
100
extended_print.tpl
100
edit_table.tpl
100
edit_buttons.tpl
100
print_button.tpl
110

Результаты подхода

- Полное отсутствие включений проверок доступа в шаблонах
- Отсутствие ошибок, связанных с доступом в период эксплуатации (6 месяцев, более 1000 пользователей, 16 групп доступа — 0 ошибок)
- Простота дальнейшей поддержки и развития (в период эксплуатации было введено в строй 2 больших модуля системы и добавлено 3 группы доступа)

Спасибо за внимание!

Кузовлев Евгений