

# Java на стороне клиента



# Неспокойно в мире 😊

1. Microsoft в Windows 8 отказалась от Silverlight;
2. Apple отказалась от Adobe Flash, заявленная в сентябре 2011 г. «поддержка» Flash в ОС iOS на деле является рендерингом Flash в видео на серверной стороне;
3. RIM выпустила новую ОС на базе QNX;
4. Позиции Google с Android усиливаются;
5. «Linux на десктопе» становится интернет-мемом, заменяющим «когда рак свистнет»



# **А некоторые говорят...**

1. Java – это медленно для «настольных» и «мобильных» приложений;
2. Java – это ресурсоёмко для «настольных» и «мобильных» приложений;
3. Java – это суровый синтаксис и скромная семантика для «настольных» и «мобильных» приложений.

**НЕ БЕЙТЕ ИХ, ОНИ ПРОСТО НЕ  
ПОНИМАЮТ...**

# Некстати...



Вопрос на \$800 миллионов:

В чём разница между торговой маркой и копирайтом?

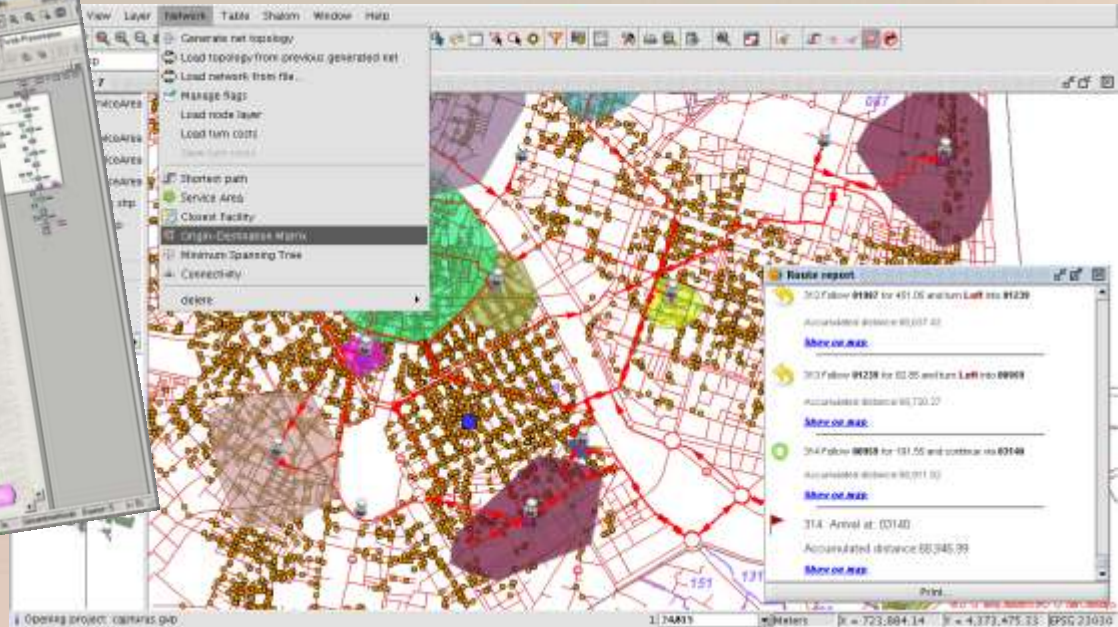
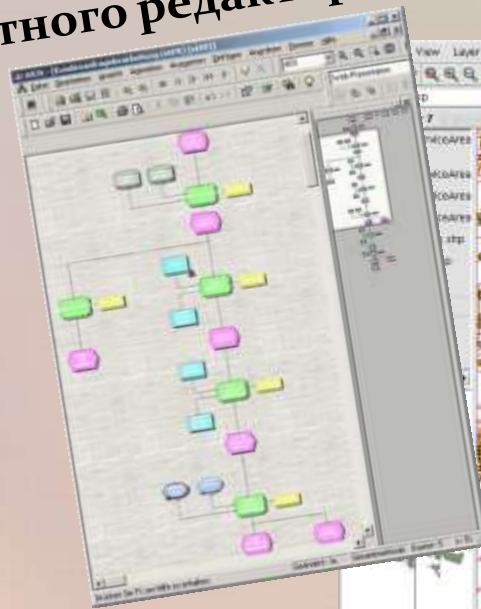
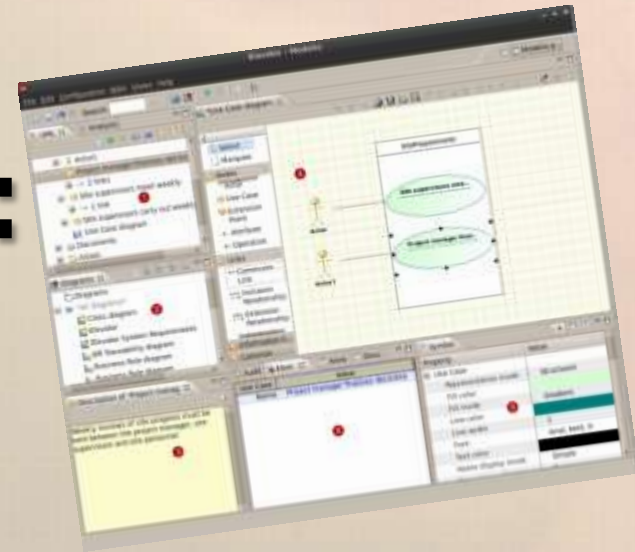
**write-once, run anywhere**

# ПОТОМУ ЧТО:

Даже монстр Modelio работает ничуть не медленнее прочих монстров:

Так IDS Sheer использует библиотеки yFiles, основу бесплатного редактора yEd:

Попробуйте сделать GIS медленной – и кто будет с ней работать ☺  
А это gvSIG Desktop:

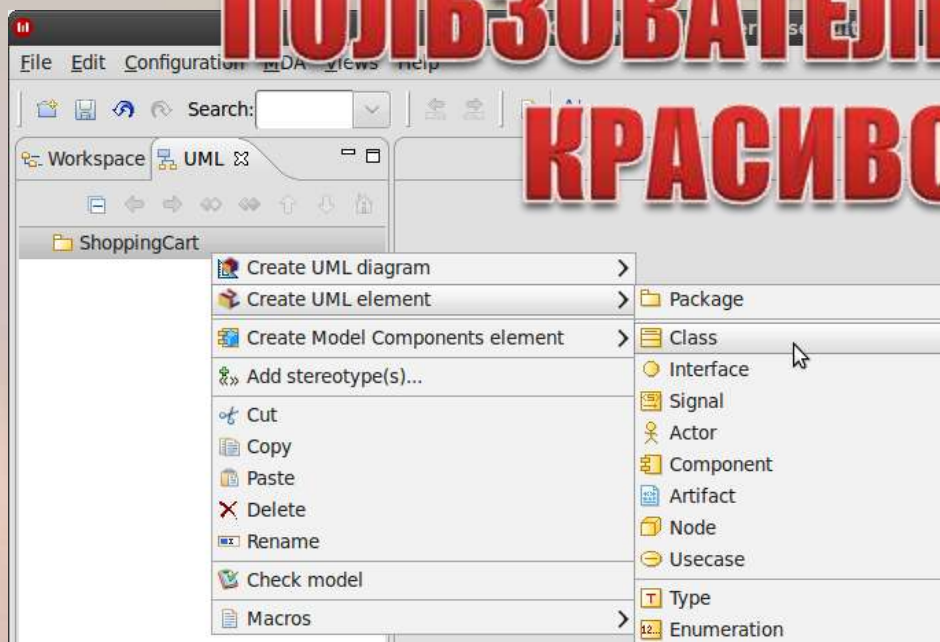


# Но есть одно «но»

Работают Java-приложения достойно, но...

**ОНИ СТРАШНЕНЬКИЕ ДЛЯ 2011 ГОДА** ☹️

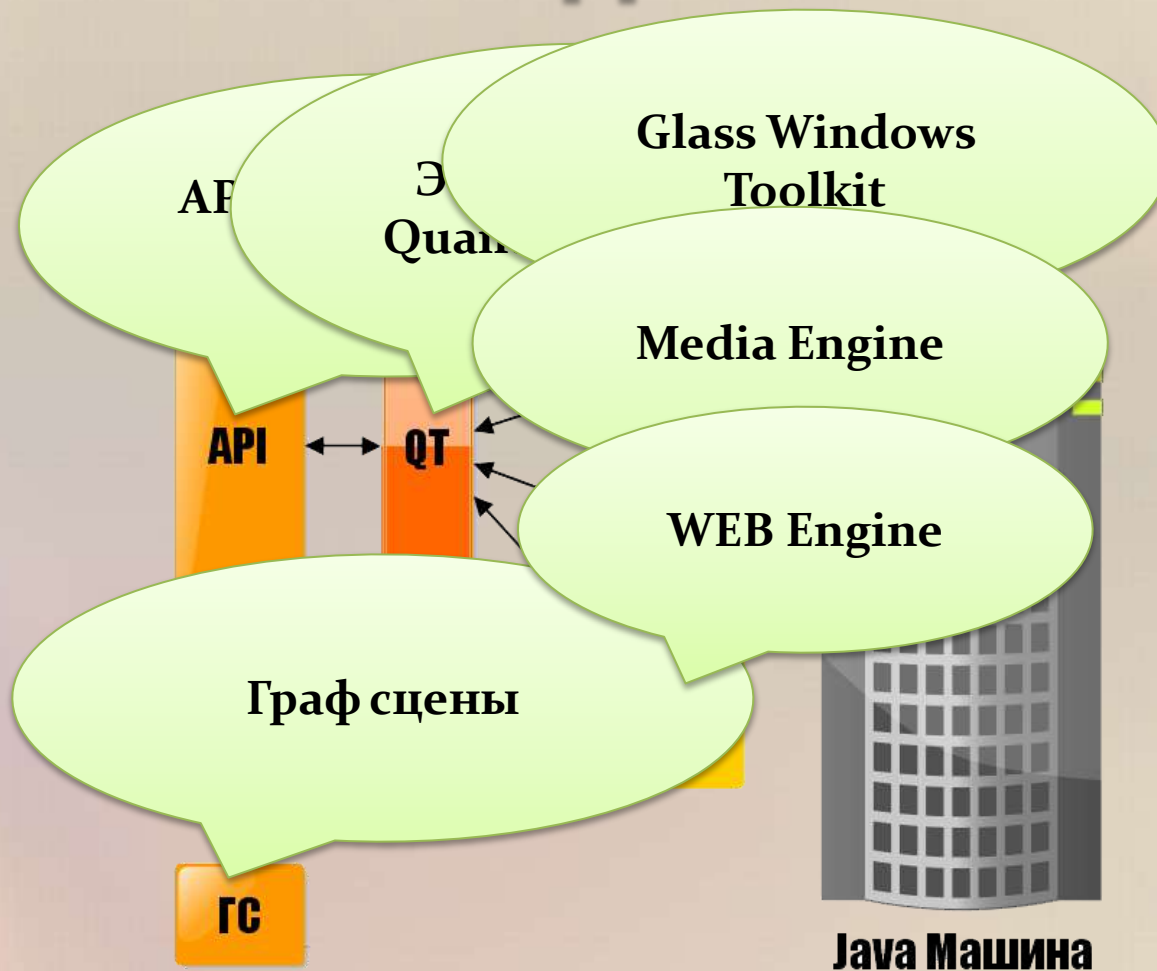
**ПОЛЬЗОВАТЕЛЬ ХОЧЕТ  
КРАСИВО!**



# И JavaFX делает красиво



# ЧЕМ ДЕЛАЕТ?



# А теперь будет **страшно**

Это же почти JavaBeans  
для «настоельных»  
программистов 2011 года?!



```
import javafx.beans.property.DoubleProperty;  
import javafx.beans.property.SimpleDoubleProperty;
```

```
class Bill
```

**Возврат property как объекта.  
Большая проблема Java.  
Хорошо бы увидеть в Java 8  
штатную поддержку  
полноценных properties!**

```
// Define a property for the amount due  
public final DoubleProperty amountDueProperty() {return amountDue;}  
}
```

```
// Define a getter for the property itself  
public DoubleProperty amountDueProperty() {return amountDue;}  
}
```

```
}
```

**ИЩЕТ!**

## Связывание? Зачем?

Понятный псевдокод:

```
class DisplayWeather
{
    window_position x = 10;
    window_position y = 20;
    forecast weather =
    bind(weather.provided);
}
```

bind гарантирует, что изменения в weather.provided всегда и автоматически будут записаны в переменную weather.

Связывание в JavaFX задаётся в compile time.

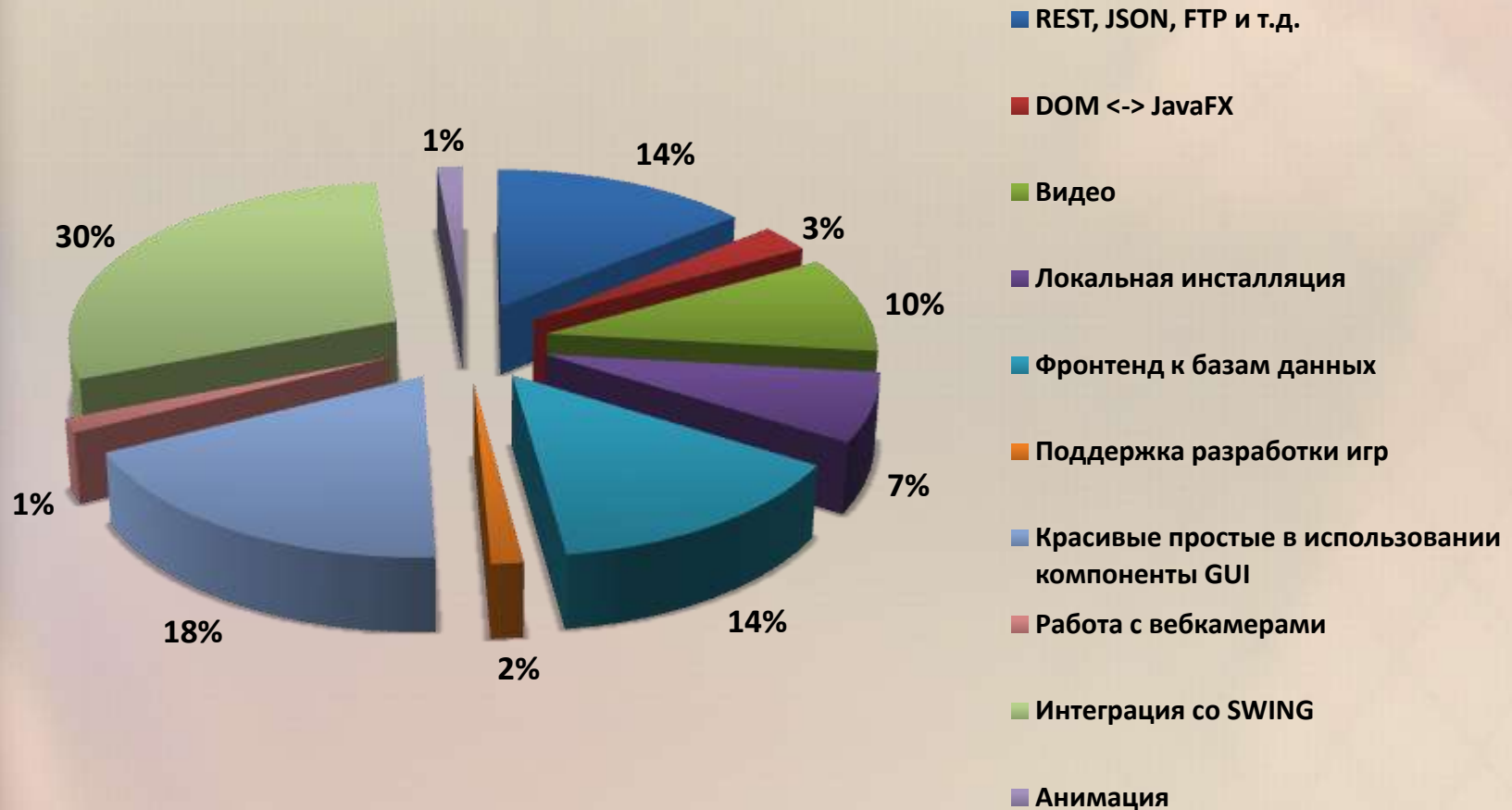
# **ПРОБЛЕМЫ JavaFX**

- 1. Турбулентное развитие**
- 2. Отсутствие killer applications**
- 3. Инструментальная поддержка**
- 4. Слабая учебная база (мы хотим много учебных приложений, и не примитивных!)**
- 5. Внутренняя конкуренция на потенциальных целевых платформах**
- 6. Неполное соответствие возможностям классических Java GUI (корпоративные клиентские приложения)**

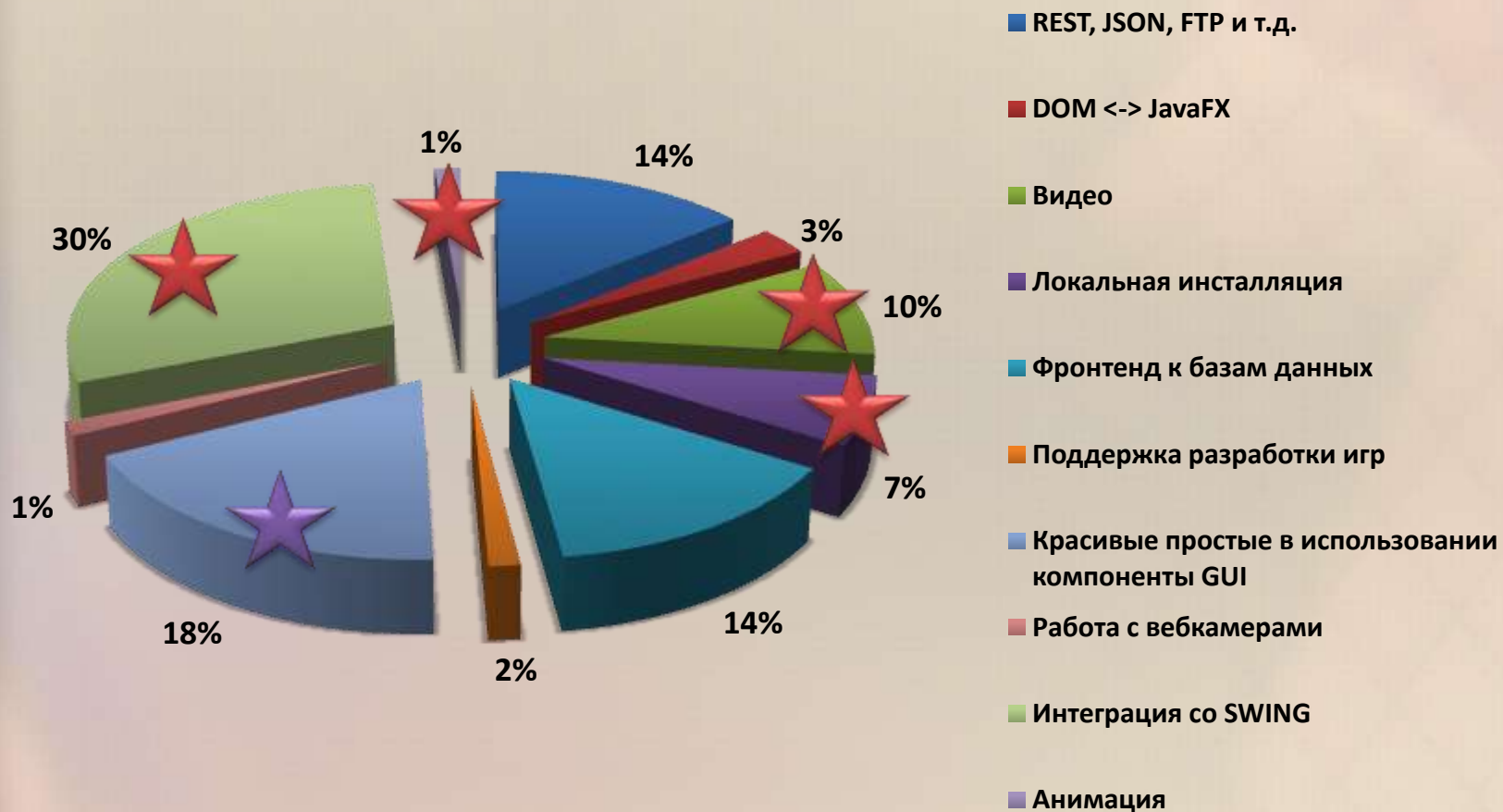
# **ЧЕГО ЖДАТЬ?**

1. **Open Source**
2. **Сверхзадачи**
3. **Превращения JavaFX в инструмент бизнеса**
4. **«Свежей крови». Специалисты в разработке корпоративных систем мягко говоря далеки от потребностей рядового некорпоративного пользователя**

# В начале 2009 года опрос программистов, использующих JavaFX, выявил требования:



# В JavaFX 2.0 реализовано:



# Ещё раз некстати...

## **write-once, run anywhere**

Типовой смартфон – экран от 3 до 5 дюймов диагональ

Типовой планшет – экран от 7 до 10 дюймов диагональ

Типовой десктоп – экран от 19 до 30 дюймов диагональ

+

Разные методы ввода и разный стиль использования устройств