

# Silverlight and Images

Looking behind the scenes

by Ivan Kirkorau, EPAM Systems

# Мысли вслух

- Silverlight – технология векторной графики.
- Silverlight работает в интернет.
- Silverlight предоставляет множество возможностей и полета для фантазии.
- Есть еще и Windows Presentation Foundation.

**Столько всего, а мы собираемся  
говорить о картинках. Зачем?**

# А они работают

- Старые фотографии



А помните что было 5-10 лет назад?



# Что от нас скрыто?

- Различные форматы изображений
- Процесс декодирования
- Преобразования цвета
- Возможности по отрисовке
- Трансформации изображений
- Прозрачность
- ...

# Начнем с WPF

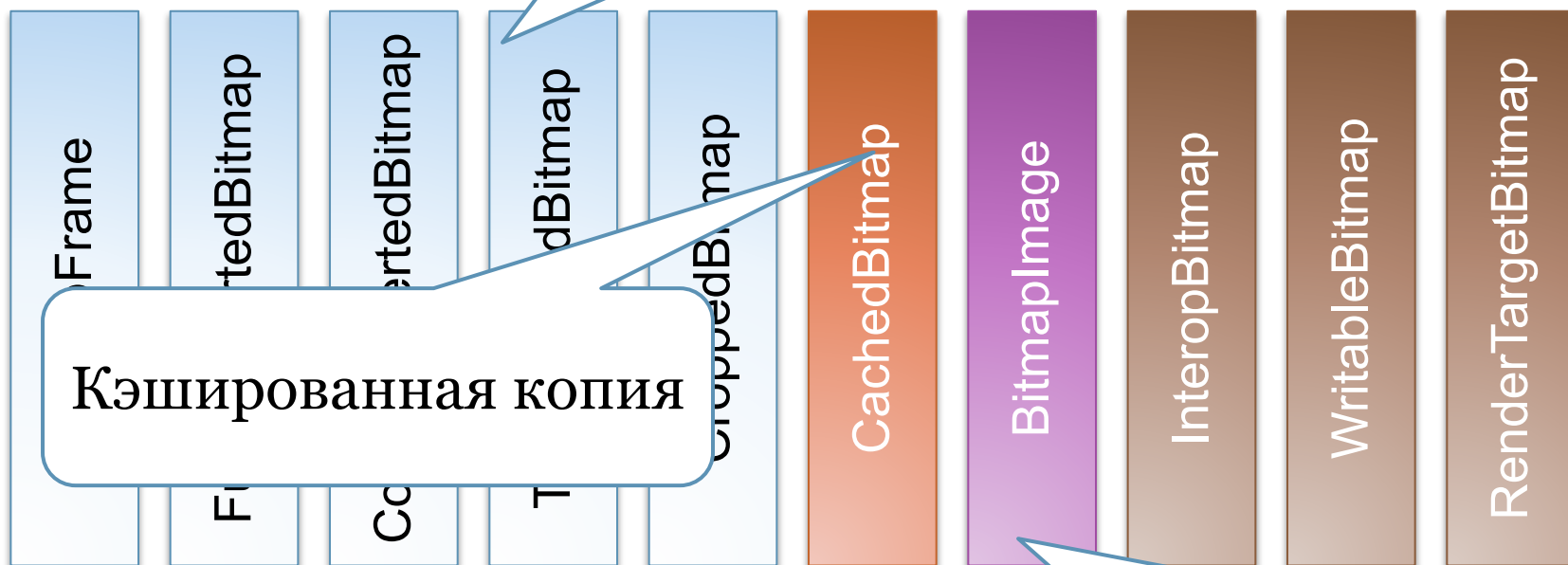
- WPF имеет полный спектр возможностей работы с изображениями.
- Silverlight – сильно ограничен.
- WPF работает в Windows и имеет доступ ко многим компонентам.
- Silverlight – в браузере и нет гарантии наличия компонентов.

# Windows Imaging Component

- WPF использует WIC
- Данный компонент инкапсулирует всю цепочку работы с изображениями
- WPF имеет оболочку над этим компонентом
- Корневой класс: **BitmapSource**
- Стандартные компоненты WIC представлены его потомками
- WPF умеет отрисовывать 2 формата цвета: Bgr32 и PArgb32 .

# Система кл

Сохраненные  
трансформации



Кэшированная копия

Основной источник  
Bitmap-ов







BitmapSource



# ISupportInitialize

- Run BeginInit
- Initialize Bitmap, run EndInit
- No more changes possible!

## Methods

	Name	Description
  	<code>BeginInit</code>	Signals the object that initialization is starting.
  	<code>EndInit</code>	Signals the object that initialization is complete.

# Dynamic images error in .NET 3.5

```
<Image>
  <Image.Source>
    <TransformedBitmap>
      <TransformedBitmap.Source>
        <BitmapImage
          UriSource="http://www.microsoft.com/.../IE.jpg"/>
        </TransformedBitmap.Source>
      <TransformedBitmap.Transform>
        <RotateTransform Angle="90"></RotateTransform>
      </TransformedBitmap.Transform>
    </TransformedBitmap>
  </Image.Source>
</Image>
```

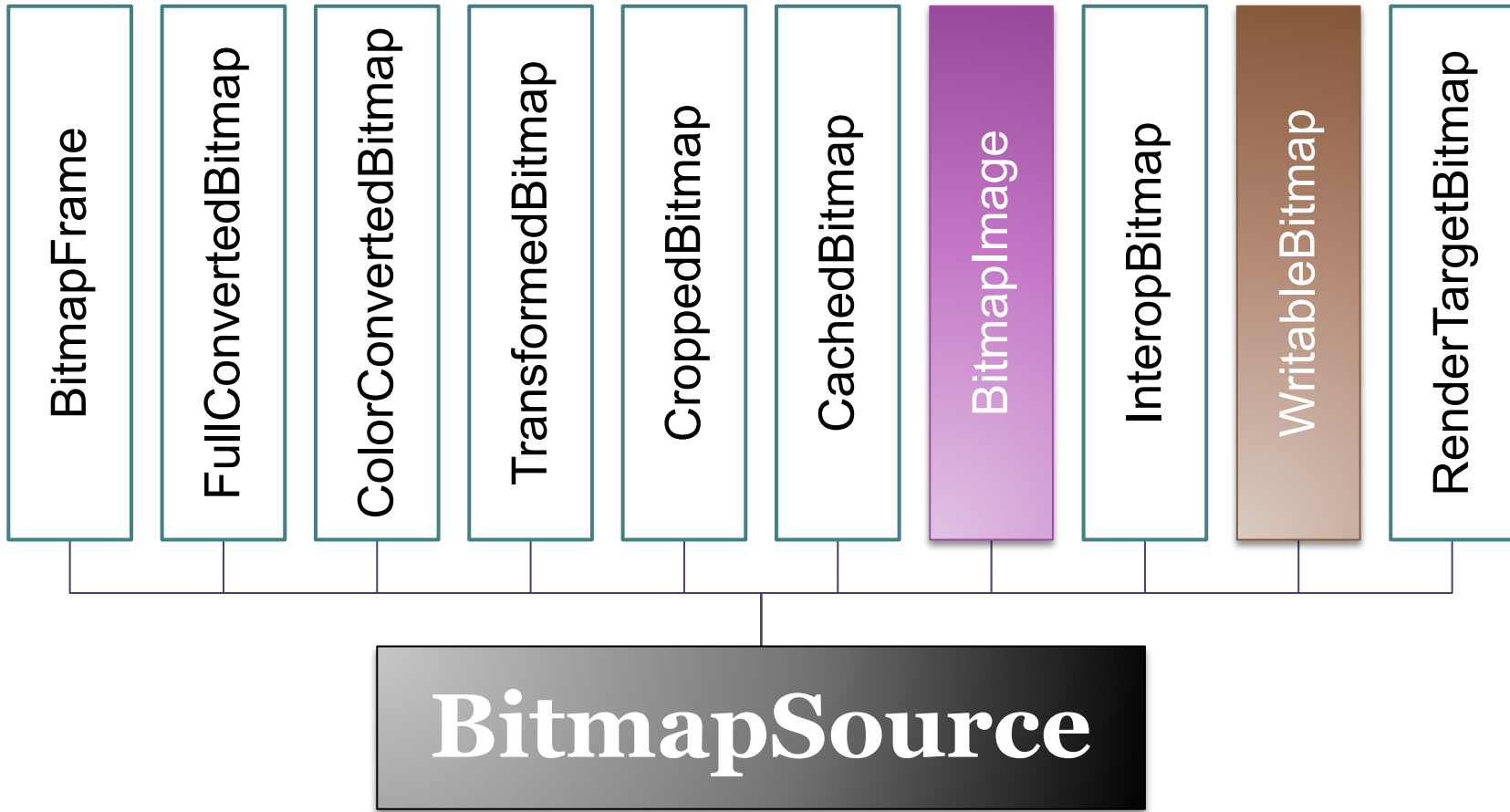
# Rendering images on screen

- Use `BitmapImage` to create `BitmapSource` from `Uri` or `Stream`.
- Classes that accept `BitmapSource`:
  - `Image`
  - `ImageBrush`
- WPF caches images for future reuse.
- `WritableImage` lets you change the contents.

# А как там Silverlight?

- Silverlight очень ограничен!
- Большая часть функциональности отсутствует.
- Любые трансформации делаются на момент рендеринга.

# A как там Silverlight?



# Выводы:

## В Silverlight:

- работа с изображениями очень ограничена.
- есть возможность работать с WriteableBitmap для создания динамических изображений.

## В WPF:

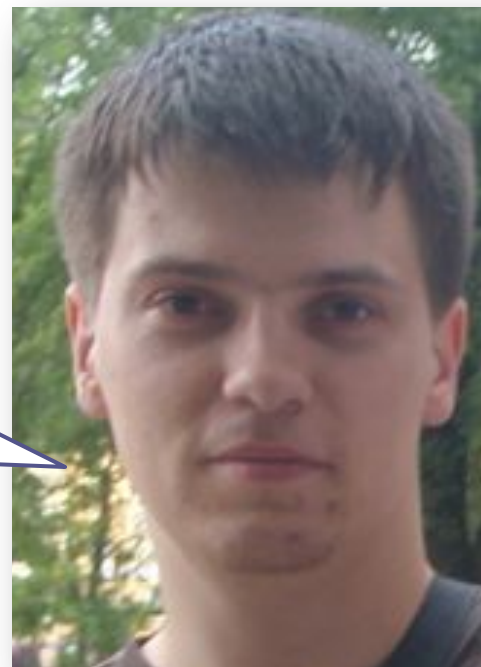
- Можно трансформировать изображения заранее, чтобы экономить ресурсы рендеринга
- Можно копировать кэшированные копии
- ...

# DEMO. Quake Light on SL4



# #devcamp Спасибо за внимание

Сейчас я отвечу  
на ваши  
вопросы.



Ivan Kirkorau  
Email: [ivan\\_kirkorau@epam.com](mailto:ivan_kirkorau@epam.com)





*The End*



The image features the text "The End" in a dark blue, elegant cursive font. The text is centered within a light blue, horizontally-oriented oval. Above and below the oval are decorative flourishes consisting of symmetrical, stylized scrollwork and floral motifs in a darker blue color. The background is white, with a dark teal horizontal bar at the top of the page.