

Projector: Your Remote IDE

Pasha Finkelshteyn, JetBrains

Did you ever complain?

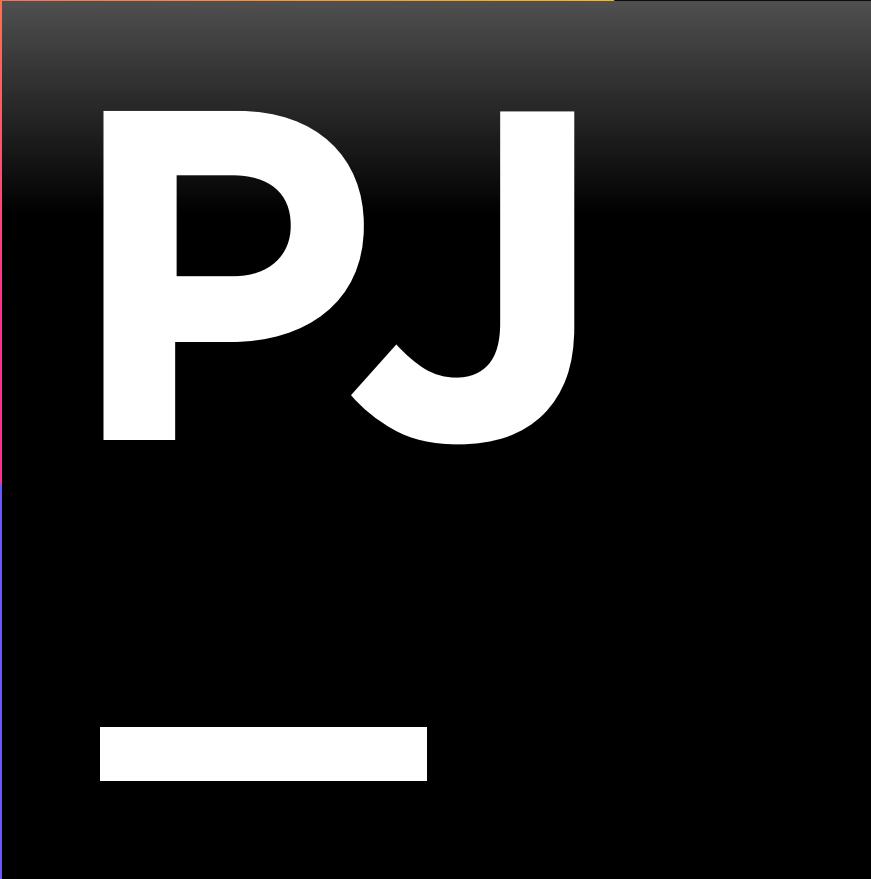
- You need to set up a VPN to access data?
- You need to use RDP to access your instance of IDEA?
- Or even VNC?
- *Long* indexing?
- Huge project compiles forever?



Did you ever want?

- Edit code from tablet/smartphone?
- Help your colleague to fix some issues?
- Show demo which accesses data from your cloud?
- Work from a weak laptop?





PJ

Issues with existing solutions. VPN

1. OpenVPN does not scale
 1. No central identity management
 2. No failover OOTB
2. Proprietary solutions are... proprietary
3. And expensive sometimes
4. IKE is not completely cross-platform

Issues with existing solutions. VNC

1. **Extremely** slow
2. Complex infrastructure (VPN?)

Issues with existing solutions. RDP

RDP is awesome

1. Single-user
2. Requires special client
3. Still needs complex infrastructure to provide access

Stop this

MARKETING

Let's dig!



Typical Swing interface

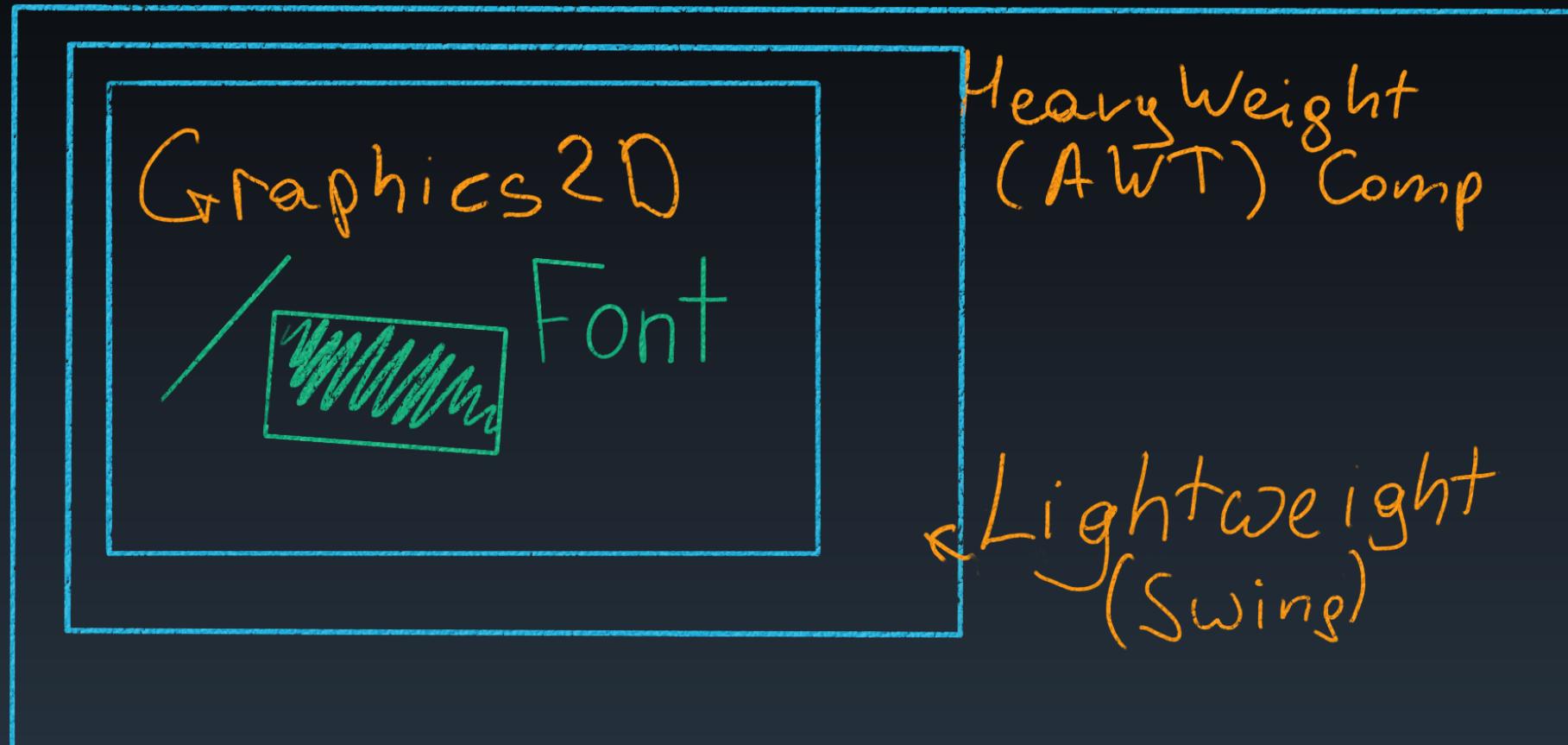


And in reality?

- Lines
- Simple shapes
- Strings
- Fonts

Swing architecture

Toolkit



Definitions

Toolkit – abstraction over the whole graphics

Heavyweight components – system-controlled components

Lightweight components – user-controlled components

Graphics2D – drawing controller

Types of drawing surfaces

- Visible
- Invisible

Part of the editor which you don't see is separate and is drawn on an invisible surface

Each window has its own surface(s)

Heavyweight components

There is a corresponding `Peer` class for each heavyweight component.

```
interface ComponentPeer {
    void paint(Graphics g);
    void setBounds(int x, int y, int width, int height, int op);
    void handleEvent(AWTEvent e);
    void setBackground(Color c);
    void setFont(Font f);
    // etc.
```

Responds for delegation to Graphics and setting several params

Heavyweight components

We have implementations of peers only for several heavyweight components.

Graphics2D

Anything may contain Graphics2D

Responds for drawing:

```
abstract class Graphics2D {  
    public abstract void drawImage(BufferedImage img,  
                                    BufferedImageOp op,  
                                    int x, int y);  
    public abstract void drawString(String str, int x, int y);  
    public abstract void fill(Shape s);  
    // etc.
```

How the hell will it work in a browser?

Welcome [DrawEventQueue](#)

Queue, that will transform any action into event, based on

```
ConcurrentLinkedQueue<List<ServerWindowEvent>>()
```



DrawEventQueue

Each component has its own queue (for now)

```
private inline fun paintArea(  
    crossinline command: DrawEventQueue.CommandBuilder.() -> Unit  
) {  
    drawEventQueue  
        .buildCommand()  
        .setClip(identitySpaceClip = identitySpaceClip)  
        .setTransform(transform.toList())  
        .setComposite(backingComposite)  
        .command()  
}
```

So we have commands. Now what?

```
while(!stopped){  
    sendAllCommands()  
    Thread.sleep(10)  
}
```

Every 10ms (roughly) aggregate all commands and send them to the client over websocket.

Format

Default format: highly optimized JSON

Supported OOTB formats: Protobuf, JSON

May support anything if browser supports it.

Backend

- Ktor?
- Tomcat?
- Undertow?
- Netty?
- Jetty?

more

GPL is not always a good thing

Popular products are not compatible with it and we HAVE TO publish source code under GPL license.

So...

[TooTallNate/Java-WebSocket](#)