

```
import React, { useState, useEffect } from 'react';

const SovereignApp = () => {
  const [input, setInput] = useState("");
  const [context, setContext] = useState("01/27/1999 Sovereign Protocol Active");
  const [isProcessing, setIsProcessing] = useState(false);
  const [timestamp, setTimestamp] = useState(new Date().toLocaleTimeString());

  // The 'Cabinet' Response State
  const [responses, setResponses] = useState({
    gemini: { text: "Standing by, Architect...", status: "idle" },
    gptProfessor: { text: "Awaiting data for audit...", status: "idle" },
    grokScout: { text: "Monitoring the X-stream...", status: "idle" }
  });

  // Live timestamp
  useEffect(() => {
    const timer = setInterval(() => {
      setTimestamp(new Date().toLocaleTimeString());
    }, 1000);
    return () => clearInterval(timer);
  }, []);

  const handleBroadcast = async () => {
    if (!input.trim()) return;

    setIsProcessing(true);
    console.log("Broadcasting to all Frontiers...");

    // Simulate staggered responses for realism
    setResponses({
      gemini: { text: "Processing directive...", status: "active" },
      gptProfessor: { text: "Analyzing request...", status: "active" },
      grokScout: { text: "Initializing scan...", status: "active" }
    });

    setTimeout(() => {
      setResponses({
        gemini: {
          text: `Interface synced. Analyzing: "${input}"\n\nMulti-modal assessment`,
          status: "complete"
        },
        gptProfessor: {
          text: `Audit initiated on directive: "${input}"\n\nLogical structure: V

```

```
        status: "complete"
    },
    grokScout: {
        text: `Kinetic pulse detected. Directive logged.\n\nScanning Georgetown
        status: "complete"
    }
}) ;
setIsProcessing(false);
}, 1500);
};

const handleKeyPress = (e) => {
    if (e.key === 'Enter' && e.ctrlKey) {
        handleBroadcast();
    }
};

const getStatusColor = (status) => {
    switch(status) {
        case 'idle': return '#666';
        case 'active': return '#ffaa00';
        case 'complete': return '#00ff41';
        default: return '#666';
    }
};

return (
    <div style={{
        backgroundColor: '#0a0a0a',
        color: '#00ff41',
        padding: '40px',
        fontFamily: '"Courier New", monospace',
        minHeight: '100vh',
        backgroundImage: 'radial-gradient(circle at 20% 50%, rgba(0, 255, 65, 0.03)
        position: 'relative'
    }}>
    /* Subtle scanline effect */
    <div style={{
        position: 'fixed',
        top: 0,
        left: 0,
        right: 0,
        bottom: 0,
        background: 'repeating-linear-gradient(0deg, transparent, transparent 2px
        pointerEvents: 'none',
        zIndex: 1
    }} />

```

```
<div style={{ position: 'relative', zIndex: 2 }}>
  /* Header */
  <div style={{ marginBottom: '30px', borderBottom: '2px solid #00ff41', padding: '10px 0' }}>
    <h1 style={{ margin: '0 0 10px 0', fontSize: '28px', letterSpacing: '2px' }}>
      AUTOGENESIS: UNIFIED GATEWAY v1.0
    </h1>
    <div style={{ display: 'flex', justifyContent: 'space-between', alignItems: 'center' }}>
      <div>
        <span style={{ color: '#00ff41' }}>ID:</span> LARRY BEDFORD III | ID:1234567890
        <span style={{ color: '#00ff41' }}> SIGNATURE:</span> 01/27/1999
      </div>
      <div style={{ color: '#00e5ff' }}>
        SYS.TIME: {timestamp}
      </div>
    </div>
  </div>
  /* Context Panel */
  <div style={{ border: '1px solid #00ff41', padding: '15px', marginBottom: '25px', backgroundColor: 'rgba(0, 255, 65, 0.05)', boxShadow: '0 0 20px rgba(0, 255, 65, 0.1)' }}>
    <h3 style={{ margin: '0 0 10px 0', fontSize: '14px', letterSpacing: '1px' }}>[SOVEREIGN CONTEXT]</h3>
    <p style={{ margin: 0, fontSize: '13px' }}>{context}</p>
  </div>
  /* Input Section */
  <div style={{ marginBottom: '30px' }}>
    <label style={{ display: 'block', marginBottom: '10px', fontSize: '12px' }}>
      DIRECTIVE INPUT → CABINET BROADCAST
    </label>
    <textarea
      style={{ width: '100%', height: '120px', backgroundColor: '#0f0f0f', color: '#ffff', border: '1px solid #333', padding: '15px', fontFamily: '"Courier New", monospace', fontSize: '14px', resize: 'none' }}>
    </textarea>
  </div>
</div>
```

```

        resize: 'vertical',
        outline: 'none',
        transition: 'border-color 0.3s, box-shadow 0.3s',
        boxShadow: isProcessing ? '0 0 15px rgba(0, 255, 65, 0.3)' : 'none'
    }
    placeholder="Enter Directive for the Cabinet... (Ctrl+Enter to broadcast)"
    value={input}
    onChange={(e) => setInput(e.target.value)}
    onKeyPress={handleKeyPress}
    disabled={isProcessing}
    onFocus={(e) => e.target.style.borderColor = '#00ff41'}
    onBlur={(e) => e.target.style.borderColor = '#333'}
/>

```

```

<button
    onClick={handleBroadcast}
    disabled={isProcessing || !input.trim()}
    style={{
        marginTop: '15px',
        padding: '12px 30px',
        cursor: isProcessing || !input.trim() ? 'not-allowed' : 'pointer',
        backgroundColor: isProcessing ? '#555' : '#00ff41',
        fontWeight: 'bold',
        border: 'none',
        color: '#0a0a0a',
        fontFamily: '"Courier New", monospace',
        fontSize: '14px',
        letterSpacing: '1px',
        transition: 'all 0.3s',
        opacity: isProcessing || !input.trim() ? 0.5 : 1,
        transform: isProcessing ? 'scale(0.98)' : 'scale(1)'
    }}
>
    {isProcessing ? '⌚ BROADCASTING...' : '▶ BROADCAST TO CABINET'}
</button>
</div>

/* Cabinet Response Grid */
<div style={{
    display: 'grid',
    gridTemplateColumns: 'repeat(auto-fit, minmax(300px, 1fr))',
    gap: '20px',
    marginTop: '30px'
}}>
    /* GEMINI */
    <div style={{
        border: `2px solid ${getStatusColor(responses.gemini.status)}`,

```

```
padding: '20px',
backgroundColor: '#0f0f0f',
transition: 'all 0.5s',
boxShadow: responses.gemini.status === 'complete' ? '0 0 20px rgba(0,
} }>
<div style={{ display: 'flex', justifyContent: 'space-between', align
  <h4 style={{ color: '#00e5ff', margin: 0, fontSize: '16px' }}>GEMIN
  <div style={{{
    fontSize: '10px',
    color: getStatusColor(responses.gemini.status),
    padding: '3px 8px',
    border: `1px solid ${getStatusColor(responses.gemini.status)}`,
    borderRadius: '3px'
  }}>
    {responses.gemini.status.toUpperCase()}
  </div>
</div>
<div style={{ fontSize: '11px', color: '#666', marginBottom: '10px' }
  <p style={{ color: '#fff', margin: 0, fontSize: '13px', lineHeight: '1
    {responses.gemini.text}
  </p>
</div>

/* GPT */
<div style={{{
  border: `2px solid ${getStatusColor(responses.gptProfessor.status)}`,
  padding: '20px',
  backgroundColor: '#0f0f0f',
  transition: 'all 0.5s',
  boxShadow: responses.gptProfessor.status === 'complete' ? '0 0 20px r
} }>
<div style={{ display: 'flex', justifyContent: 'space-between', align
  <h4 style={{ color: '#ff00ff', margin: 0, fontSize: '16px' }}>GPT</
  <div style={{{
    fontSize: '10px',
    color: getStatusColor(responses.gptProfessor.status),
    padding: '3px 8px',
    border: `1px solid ${getStatusColor(responses.gptProfessor.status)}`,
    borderRadius: '3px'
  }}>
    {responses.gptProfessor.status.toUpperCase()}
  </div>
</div>
<div style={{ fontSize: '11px', color: '#666', marginBottom: '10px' }
  <p style={{ color: '#fff', margin: 0, fontSize: '13px', lineHeight: '1
    {responses.gptProfessor.text}
  </p>
```

```

        </div>

        {/* GROK */}
        <div style={{
            border: `2px solid ${getStatusColor(responses.grokScout.status)}`,
            padding: '20px',
            backgroundColor: '#0f0f0f',
            transition: 'all 0.5s',
            boxShadow: responses.grokScout.status === 'complete' ? '0 0 20px rgba
        }}>
            <div style={{ display: 'flex', justifyContent: 'space-between', align
                <h4 style={{ color: '#ffff00', margin: 0, fontSize: '16px' }}>GROK<
                <div style={{
                    fontSize: '10px',
                    color: getStatusColor(responses.grokScout.status),
                    padding: '3px 8px',
                    border: `1px solid ${getStatusColor(responses.grokScout.status)}`,
                    borderRadius: '3px'
                }}>
                    {responses.grokScout.status.toUpperCase()}
                </div>
            </div>
            <div style={{ fontSize: '11px', color: '#666', marginBottom: '10px' }>
                <p style={{ color: '#fff', margin: 0, fontSize: '13px', lineHeight: '1.5', textOverflow: 'ellipsis', whiteSpace: 'pre-line' }}>
                    {responses.grokScout.text}
                </p>
            </div>
        </div>

        {/* Footer */}
        <div style={{
            marginTop: '40px',
            paddingTop: '20px',
            borderTop: '1px solid #333',
            fontSize: '11px',
            color: '#555',
            textAlign: 'center'
        }}>
            BILLION-NODE ARCHITECTURE | MULTI-FRONTIER INTEGRATION | SOVEREIGN PROT
        </div>
    </div>
</div>
);

};

export default SovereignApp;

```

