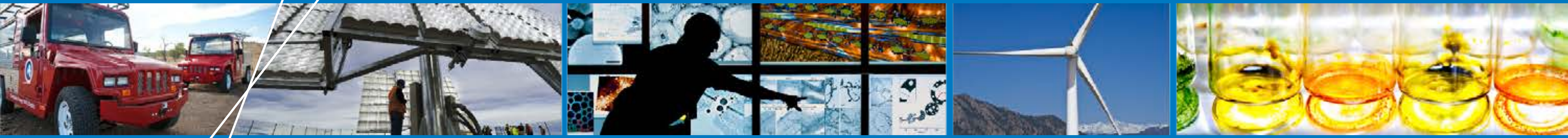
Introduction to the SAM Software Development Kit (SDK)



SAM Webinar

Paul Gilman

July 24, 2013

NREL is a national laboratory of the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, operated by the Alliance for Sustainable Energy, LLC.

Call-in numbers

- USA 1-203-607-0666 877-951-7311
- Audience passcode: 4698503

Outline

- What is the SAM SDK?
- Examples of how people are using the SDK
- Overview of SDK contents
- Simple example showing the SDK in action

What is the SAM SDK?

- **Alphabet soup!**

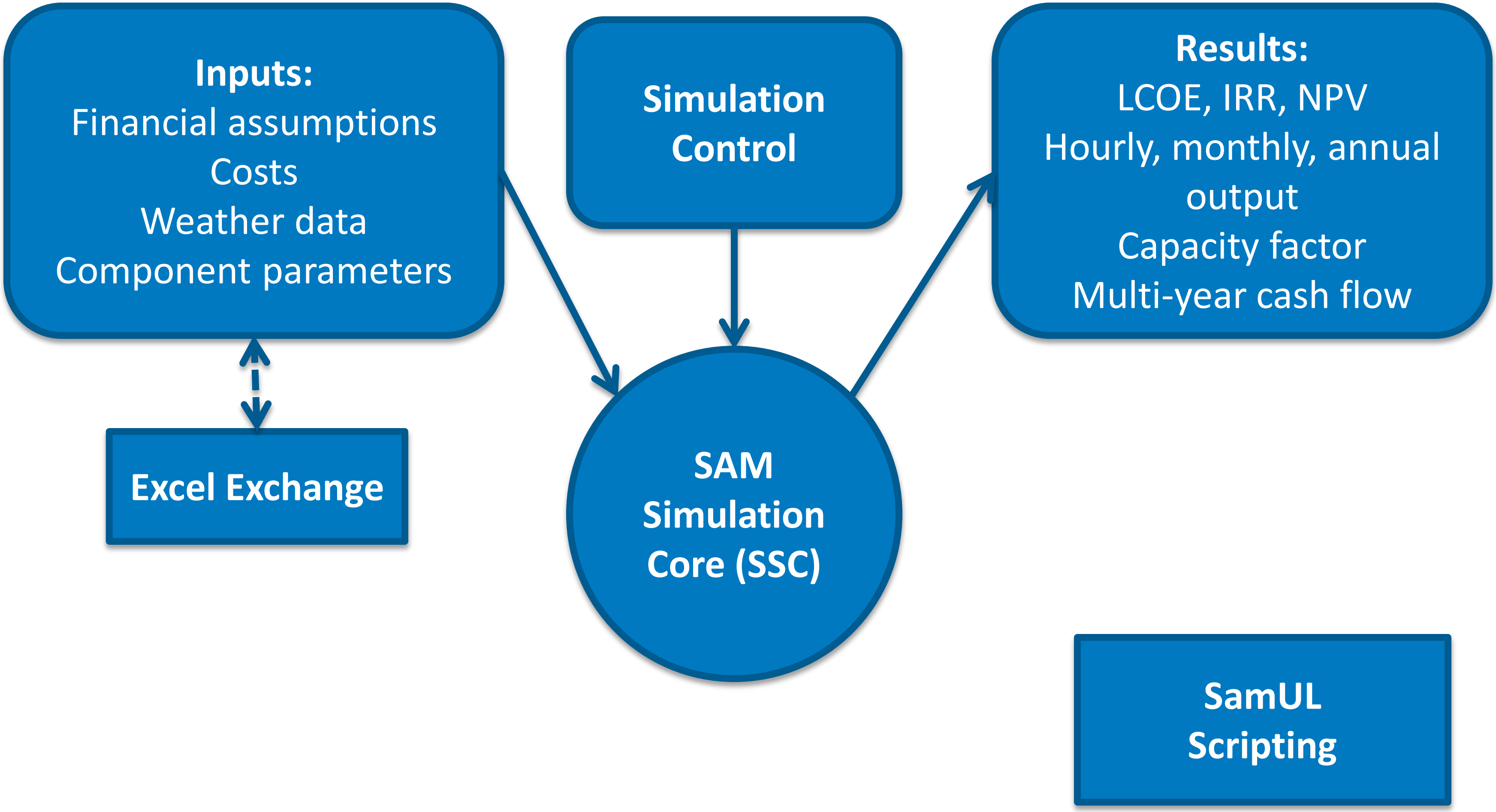
SAM	System Advisor Model
SDK	Software development kit
API	Application programming interface
SSC	SAM simulation core
SSCdev	SSC development tool
LK	Language kit, name of the SSCdev scripting language

What is the SAM SDK?

- **A set of tools that allows you to access the SAM Application Programming Interface (API) from your own programs**
  - Run simulation modules to model a renewable energy component, system or project
  - Set inputs, run simulations, read results
  - Automate repetitive modeling tasks
  - Design your own user interface for desktop or web applications

The SDK does not allow you to modify a simulation module’s source code

The SSC and SAM’s user interface



Inputs:
Financial assumptions
Costs
Weather data
Component parameters

Simulation Control

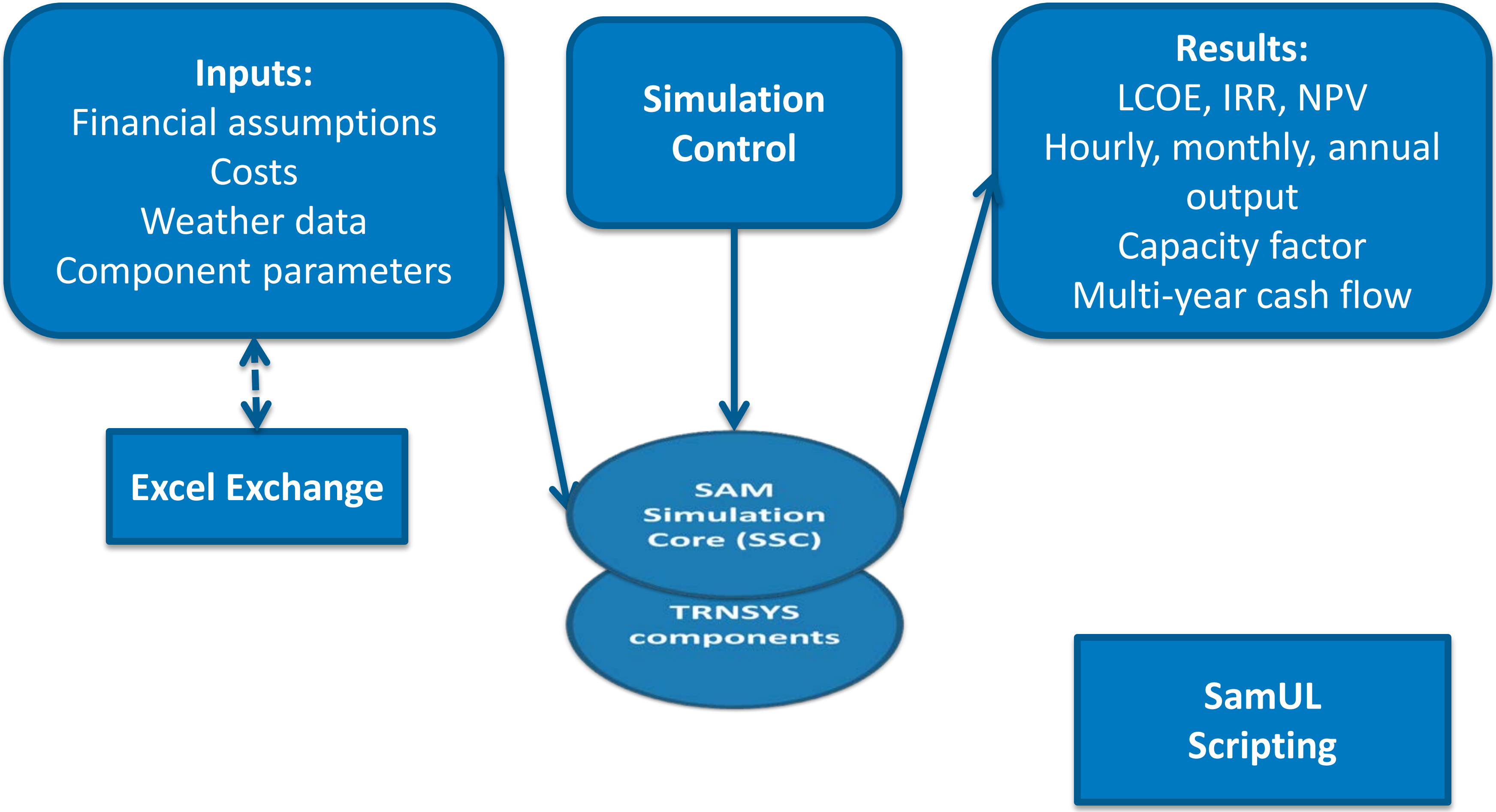
Results:
LCOE, IRR, NPV
Hourly, monthly, annual output
Capacity factor
Multi-year cash flow

Excel Exchange

SAM Simulation Core (SSC)

SamUL Scripting

The SSC and SAM's user interface



Inputs:
Financial assumptions
Costs
Weather data
Component parameters

Simulation Control

Results:
LCOE, IRR, NPV
Hourly, monthly, annual output
Capacity factor
Multi-year cash flow

Excel Exchange

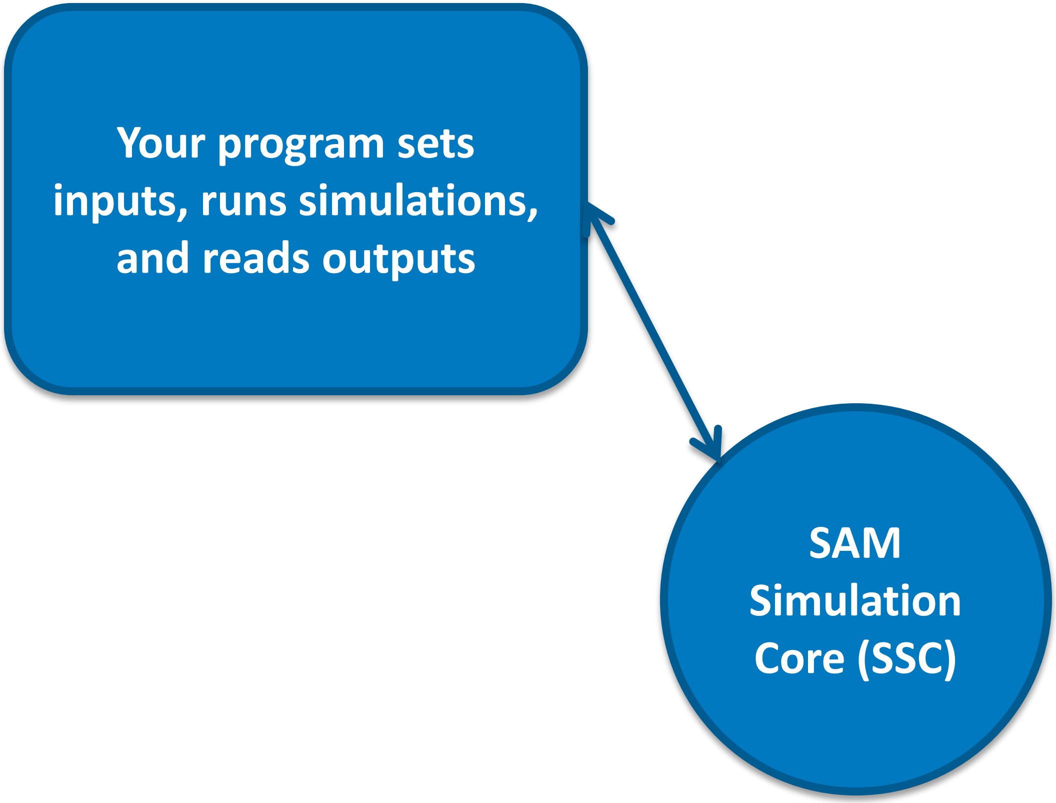
SAM Simulation Core (SSC)

TRNSYS components

SamUL Scripting

CSP model components are not available in the SSC as of July 2013, but will be in future versions of the SDK

The SSC API and your programs


Your program sets inputs, runs simulations, and reads outputs
SAM Simulation Core (SSC)

SSC terminology

- **Module (simulation module, compute module)**
  - Chunk of compiled code that simulates a system component, project cash flow, or performs a modeling function
  - Your code combines modules to build a system or project model
- **Data container**
  - Group of variables used by the modules

SSC modules as of July 2013

Photovoltaic Model Components

6parsolve	Calculates a set of parameters from manufacturer module specifications for the CEC module model
pv6parmod	CEC module model
pvsandiainv	Sandia inverter model
pvsamv1	Flat-plate PV model
pvwattsfunc	Runs a the PVWatts model for a single timestep instead of as an 8760 hourly model
pvwattsv1	PVWatts model

Wind Model

windpower	Wind power model

Geothermal Model Components

geothermal	Geothermal power model
geothermalui	Calculates interim design values for the geothermal power model

Financial Models and Utilities

annualoutput	Calculates Year 1 hourly and monthly, and Years 2 and later total annual system output for use in cash flow calculations
cashloan	Residential and commercial financial model
utilityrate (1 and 2)	Calculates the value of energy using complex utility rate structures for Versions 1 and 2 of the utility rate database
equpartflip	Partnership flip financial models
ippppa	Commercial PPA and Independent Power Producer (IPP) financial models
levpartflip	Leveraged partnership flip financial model
saleleaseback	Sale lease back financial model
singleowner	Single owner financial model

Solar Hot Water Model

swh	Solar water heating model

Weather File Modules

wfreader	Reads format weather data files in the following formats: TMY2, TMY3, EPW, SMW)
irradproc	General purpose irradiance processor for calculating POA

Test Modules

test_irr	Internal rate of return calculator from utility financing models for testing
test_pvshade	PV self-shading model for testing

Utility Modules

timeseq	Converts start time, end time, and time step into arrays of time stamps (seconds, minutes, hours, days, months)

SSC modules for a residential PV project model

- pvsamv1
- annualoutput
- utilityrate
- cashloan

Examples of Applications using the SSC API

- **SunRun**
  - PVWatts and flat plate PV models for internal pricing and system design tools
- **SunEdison**
  - PVWatts for energy estimates
- **Arizona Public Service**
  - Single-timestep PVWatts function for grid-integration analysis
- **Concept3D**
  - Flat plate PV model and residential/commercial financial model in simuwatt
- **Genability**
  - PVWatts for online siting tool

What is in the SDK?

- **The native C API: sscapi.h**
- **Documentation**
  - SSC Guide
  - Code examples
  - Comments in the code examples and API
- **SSC development tool: SSCdev**
  - Module and variable browsers
  - LK script for building and testing models
  - LK Guide
- **SSC libraries for Windows, OS X, Linux**
- **Language wrappers for C#, Java, MATLAB, Python**

Getting started with the SDK

Implement your model in your development environment

Getting started with the SDK

1. **Read the SSC guide**
2. **Set up your development environment in C, C#, Java, MATLAB, or Python**
   - Read any ReadMe files and comments in the code
   - Run the test programs provided in the Languages folder
3. **Build and test your model in SSCdev**
4. **Implement your model in your development environment**

Working with the SDK requires trial and error

http://kingmagic.wordpress.com/2006/11/25/trial-error/